

UILU-ENG-95-2560

PHASOR REPRESENTATION IN POWER SYSTEM DYNAMIC MODELING

MICHAEL PATRICK RANSICK

Power Affiliates Program

Department of Electrical and Computer Engineering

University of Illinois at Urbana-Champaign

Urbana, Illinois 61801

PAP-TR-95-10

January 1996

FOREWARD

This technical report is a reprint of the thesis written by Michael Patrick Ransick as partial fulfillment of the requirements for the degree of Master of Science in Electrical and Computer Engineering at the University of Illinois.

Peter Sauer

Thesis Advisor

January 1996

TABLE OF CONTENTS

	Page
1. INTRODUCTION.....	1
2. MODELING.....	2
2.1 Full Model.....	2
2.2 Reduced Model.....	5
2.3 The dq to abc Transformation.....	5
2.4 The dq Phasors.....	5
3. PHASOR MEASUREMENTS.....	6
3.1 The Discrete Fourier Transform.....	6
3.2 The DFT Properties.....	7
3.2.1 Frequency response.....	7
3.2.2 Aliasing.....	11
3.2.3 Picket-fence effect.....	11
3.2.4 Leakage.....	12
4. SIMULATION.....	13
4.1 Generator Data.....	13
4.2 Full Model Results.....	14
4.2.1 Full model phasors.....	15
4.2.2 DFT error due to frequency.....	16
4.2.3 The dc phasor.....	17
4.3 Reduced Model Results.....	17
4.3.1 Reduced model phasors.....	19
4.3.2 The dc phasor.....	21
4.4 Half-Cycle DFT Results.....	22
4.5 The dc Component Rejection in One-Cycle DFT.....	27
4.6 Phasor Comparison.....	29
5. CONCLUSIONS.....	30
REFERENCES.....	31
APPENDIX A. FORTRAN CODE.....	32
A.1 Full Model.....	32
A.2 Reduced Model.....	41
A.3 Half-Cycle Phasor Subroutine.....	51

1. INTRODUCTION

Advances in computing power and time synchronization offer new opportunities in analysis, simulation, and control of power systems. The math intensive task of computing phasors was at one time only practical in an off-line analysis or simulation mode. With the cost of microprocessors plummeting and the availability of memory increasing, real-time phasor measurements are now practical. Real-time phasor measurements coupled with the now common availability of time synchronization signals allow the precise state of a power system to be monitored in real time [1]. The real-time power system measurements can be used to control power flow [2] and possibly correct power swings before service is lost or equipment is damaged. To effectively use real-time phasor measurements, the user must understand how the phasors relate to the traditional d-q phasors obtained by system simulation. This thesis explores the similarities and differences between real-time phasor measurements and the traditional d-q magnitude and phase obtained by simulation and focuses on a single synchronous machine with a balanced three-phase resistive load and balanced three-phase resistive fault for the initial explanations of real-time phasors.

2. MODELING

2.1 Full Model

The model used for simulation is presented here without derivation. The model used has transformed the system from the a-b-c to the d-q-o reference frame and scaled all variables to be in per-unit values. The following model is well-accepted as a representation of the dynamics associated with a single generator serving a resistive load [3]:

$$\frac{1}{w_s} \frac{d\psi_d}{dt} = R_s I_d + \frac{w}{w_s} \psi_q + V_d \quad (1)$$

$$\frac{1}{w_s} \frac{d\psi_q}{dt} = R_s I_q - \frac{w}{w_s} \psi_d + V_q \quad (2)$$

$$\frac{d\delta}{dt} = w - w_s \quad (3)$$

$$\frac{2H}{w_s} \frac{dw}{dt} = T_M - \psi_d I_q + \psi_q I_d \quad (4)$$

$$T_{do}' \frac{dE_q'}{dt} = -E_q' - (X_d - X_d') [I_d - \frac{(X_d' - X_d'')}{(X_d' - X_{ls})^2} (\psi_{1d} + (X_d' - X_{ls}) I_d - E_q')] + E_{fd} \quad (5)$$

$$T_{do}'' \frac{d\psi_{1d}}{dt} = -\psi_{1d} + E_q' - (X_d' - X_{ls}) I_d \quad (6)$$

$$T_{qo}' \frac{dE_d'}{dt} = -E_d' + (X_q - X_q') [I_q - \frac{(X_q' - X_q'')}{(X_q' - X_{ls})^2} (\psi_{2q} + (X_q' - X_{ls}) I_q - E_d')] \quad (7)$$

$$T_{qo}'' \frac{d\psi_{2q}}{dt} = -\psi_{2q} - E_d' - (X_q' - X_{ls}) I_q \quad (8)$$

$$T_E \frac{dE_{fd}}{dt} = -(K_E + S_E) E_{fd} + V_R \quad (9)$$

$$T_F \frac{dR_f}{dt} = -R_f + \frac{K_F}{T_F} E_{fd} \quad (10)$$

$$T_A \frac{dV_R}{dt} = -V_R + K_A R_f - \left(\frac{K_A K_F}{T_F} \right) E_{fd} + K_A (V_{ref} - V_t) \quad (11)$$

$$T_{CH} \frac{dT_M}{dt} = -T_M + P_{SV} \quad (12)$$

$$T_{SV} \frac{dP_{SV}}{dt} = -P_{SV} + P_C - \frac{1}{R_d} \left(\frac{w}{w_s} \right) \quad (13)$$

With algebraic equations for the currents, terminal constraints, and limit constraints,

$$I_d = \frac{1}{X_d''} (-\psi_d + \frac{(X_d'' - X_{ls})}{(X_d' - X_{ls})} E_q' + \frac{(X_d' - X_d'')}{(X_d' - X_{ls})} \psi_{1d}) \quad (14)$$

$$I_q = \frac{1}{X_q''} (-\psi_q - \frac{(X_q'' - X_{ls})}{(X_q' - X_{ls})} E_d' + \frac{(X_q' - X_q'')}{(X_q' - X_{ls})} \psi_{2q}) \quad (15)$$

$$V_d = I_d R_{load} \quad (16)$$

$$V_q = I_q R_{load} \quad (17)$$

$$V_t = \sqrt{V_d^2 + V_q^2} \quad (18)$$

$$V_{R_{min}} \leq V_R \leq V_{R_{max}} \quad (19)$$

$$0 \leq P_{SV} \leq P_{SV_{max}} \quad (20)$$

The states are

w is the rotor speed.

ψ_d and ψ_q are the scaled stator d and q axis fluxes.

δ is the rotor angle.

E'_d, E'_q, ψ_{1d} and ψ_{2q} are the scaled rotor fluxes.

E_{fd} is the output of the main exciter.

V_R is the output of the voltage regulator.

R_f is the system stabilizer feedback.

P_{SV} is the position of the steam valve.

T_M is the shaft mechanical torque.

The d and q axis variables are

I_d and I_q are the scaled stator d and q axis currents.

V_d and V_q are the scaled stator d and q axis voltages.

It is important to note that the preceding model was developed with four general assumptions, which are [4]

- a. Stator has three coils in a balanced symmetrical configuration centered 120 electrical degrees apart.
- b. Rotor has four coils in a balanced symmetrical configuration located in pairs 90 electrical degrees apart.
- c. The relationship between the flux linkages and currents must reflect a conservative coupling field.
- d. The relationship between the flux linkages and currents must be independent of Θ_{shaft} when expressed in the d-q-o coordinate system.

2.2 Reduced Model

For reduced-order modeling the fast dynamics of the stator are neglected [4]. Then the first two differential equations of the full model (for ψ_d and ψ_q) become algebraic.

$$0 = R_s I_d + \psi_q + V_d \quad (21)$$

$$0 = R_s I_q - \psi_d + V_q \quad (22)$$

2.3 The dq to abc Transformation

The original “abc” variables can be recovered from this model through the transformation [4],

$$I_a = \sqrt{2} \sqrt{I_d^2 + I_q^2} \cos(\omega_s t + \delta + \tan^{-1} \frac{I_q}{I_d} - \frac{\pi}{2}) \quad (23)$$

$$I_b = \sqrt{2} \sqrt{I_d^2 + I_q^2} \cos(\omega_s t + \delta + \tan^{-1} \frac{I_q}{I_d} - \frac{\pi}{2} - \frac{2\pi}{3}) \quad (24)$$

$$I_c = \sqrt{2} \sqrt{I_d^2 + I_q^2} \cos(\omega_s t + \delta + \tan^{-1} \frac{I_q}{I_d} - \frac{\pi}{2} + \frac{2\pi}{3}) \quad (25)$$

2.4 The dq Phasors

The following complex variables can be defined for the models:

$$\overline{I_{DQ}} \equiv (I_D + jI_Q) \equiv (I_d + jI_q)e^{j(\delta - \pi/2)} \quad (26)$$

$$\overline{V_{DQ}} \equiv (V_D + jV_Q) \equiv (V_d + jV_q)e^{j(\delta - \pi/2)} \quad (27)$$

where I_d and I_q are the d and q coordinate currents and V_d and V_q are the d and q coordinate voltages of the model. These are referred to as the exact dynamic phasors for the full model and as the approximate dynamic phasors for the reduced-order model.

3. PHASOR MEASUREMENTS

3.1 The Discrete Fourier Transform

The fundamental component of the Discrete Fourier Transform (DFT) is defined in [2] as

$$\bar{I}_{a \text{ DFT}} \equiv \frac{\sqrt{2}}{N} \sum_{k=1}^N I_{ak} e^{-jk \frac{2\pi}{N}} \quad (28)$$

where I_{ak} is the k th sample of the phase a current, N is the number of samples, and k is the sample number. $\bar{I}_{a \text{ DFT}}$ is then the DFT of the phase a current. We will call it the measured dynamic phasor. Similarly, the n th harmonic component of the waveform is defined as

$$\bar{I}(n)_{a \text{ DFT}} \equiv \frac{\sqrt{2}}{N} \sum_{k=1}^N I_{ak} e^{-jnk \frac{2\pi}{N}} \quad (29)$$

where n is the harmonic number. By replacing n in Equation (29) with the desired harmonic component number, the phasor of any component can be measured. Unless otherwise specified, the phasors in this thesis are calculated for the fundamental component. Therefore, n is replaced by one. Similarly, if the dc component is desired, n is replaced by zero. For the second harmonic, n is replaced by two. When the fundamental component of all three phases is calculated, the following positive sequence phasor can be defined [5]:

$$\bar{I}_{POS \text{ DFT}} \equiv \frac{1}{3} (\bar{I}_{a \text{ DFT}} + \bar{I}_{b \text{ DFT}} e^{j\frac{2\pi}{3}} + \bar{I}_{c \text{ DFT}} e^{-j\frac{2\pi}{3}}) \quad (30)$$

This is referred to as the “measured” dynamic phasor. We will compute the phasor at every sample by discarding the oldest sample and adding the latest sample to the calculation. This is called a sliding DFT since it is moving with time. This allows us to look at each frequency component as it changes in time instead of looking at a snapshot of the waveform as is done in

spectral analysis. A judicious choice of sampling frequency results in computationally efficient algorithms [6].

3.2 The DFT Properties

3.2.1 Frequency response

When the DFT is calculated for one frequency component, such as the fundamental, it acts as a band-pass filter for that frequency. The response varies with the length of the data window.

Figure 1 is the traditional frequency response plot for the fundamental component. The graph is developed by transforming the DFT equation into the z domain and running a frequency response curve. This is not a true frequency response as we know it. This is the DFT response for one frequency input.

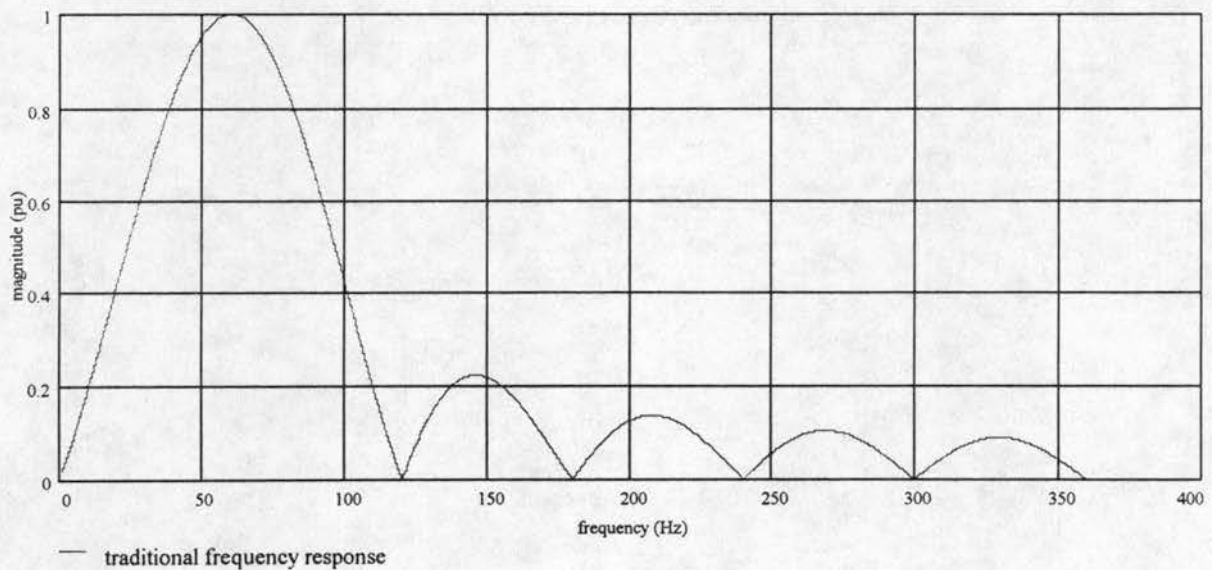


Figure 1. Traditional DFT frequency response.

When running a one-cycle DFT without windowing, the frequency response actually looks quite different. When performing a DFT on a partial cycle of a waveform, energy is spread into other

frequencies. For instance, if the input waveform is $\sin(\omega t + \Theta)$ with ω varied from 0 to 360 Hz and $\Theta = 0$, the frequency response actually has a peak of 1.024 pu versus 1.0 pu for the traditional frequency response. The frequency response is shown in Figure 2.

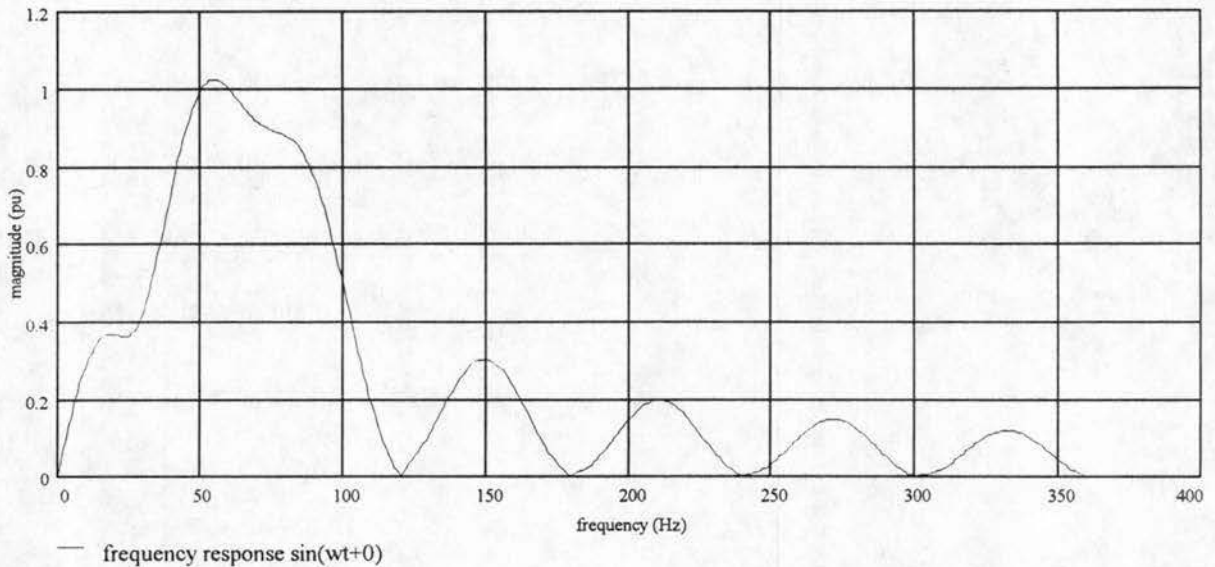


Figure 2. One cycle DFT frequency response with input = $\sin(\omega t + 0)$.

If the input waveform is $\sin(\omega t + \Theta)$ with ω varied from 0 to 360 Hz and $\Theta = 45$ degrees, the frequency response is shown in Figure 3. If the input waveform is $\sin(\omega t + \Theta)$ with ω varied from 0 to 360 Hz and $\Theta = 90$ degrees, the frequency response is shown in Figure 4. The frequency response changes when the angle of the input waveform changes. This occurs because the one cycle of data used for the DFT has discontinuities at the end points of the data window. The discontinuity changes as the phase angle changes.

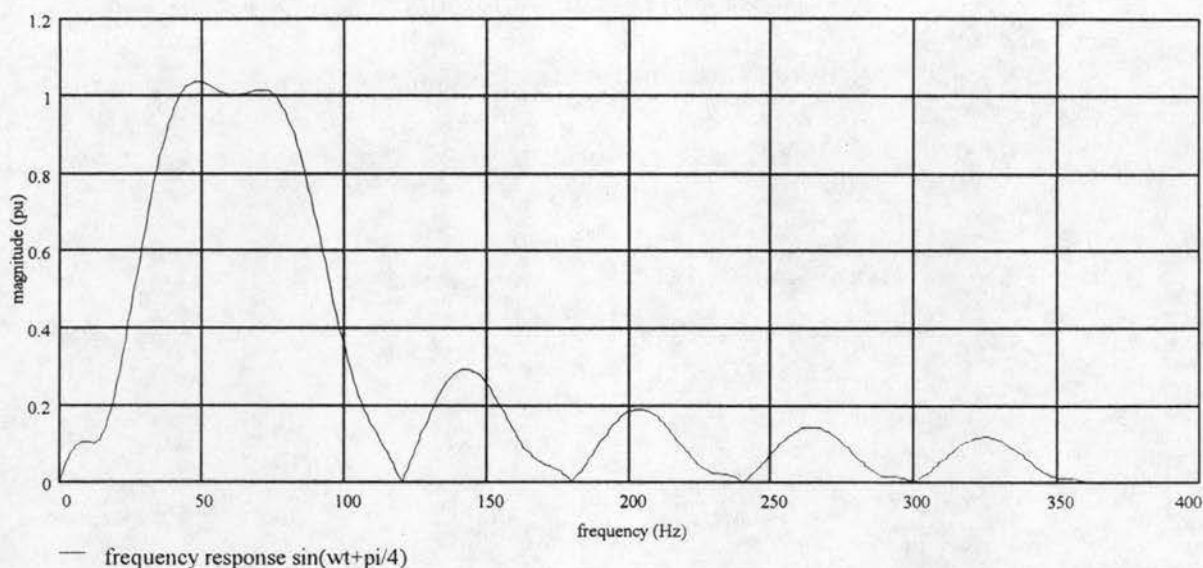


Figure 3. One cycle DFT frequency response with input = $\sin(\omega t + \pi/4)$.

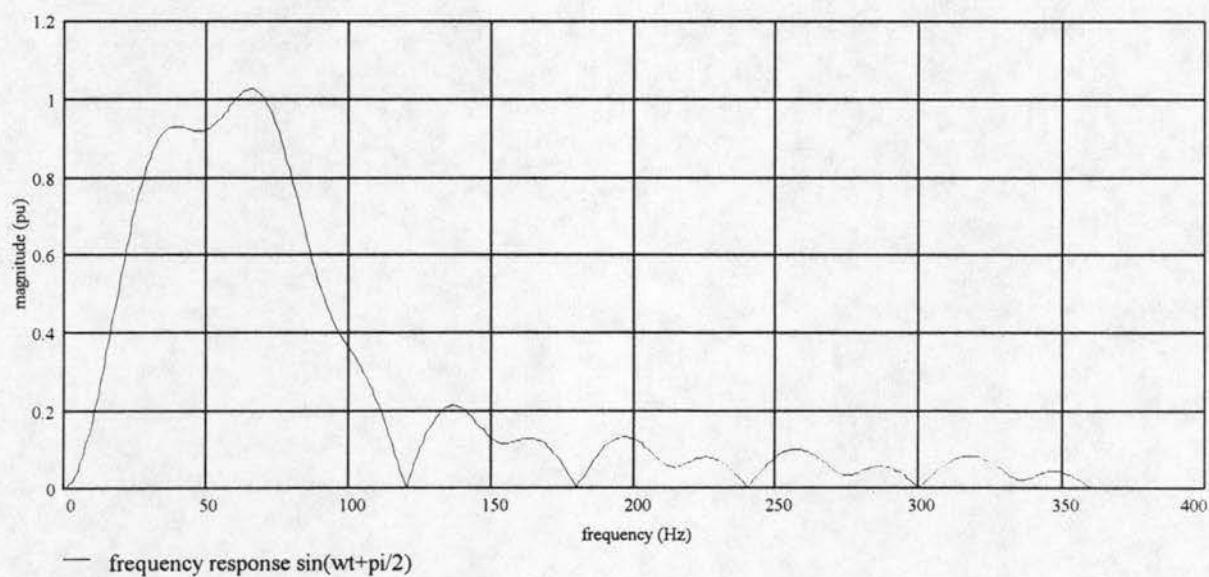


Figure 4. One cycle DFT frequency response with input = $\sin(\omega t + \pi/2)$.

In all four graphs, the magnitude of the fundamental component is one. The graphs show that if the frequency of the signal being measured is not exactly the fundamental frequency, there is an

error. Phadke and Thorp have expressed this error for the positive sequence phasor in geometric terms in [2]. (Their error assumes the frequency response in Figure 1.) The complex attenuation factor which distorts the true phasor varies with frequency. The equation is

$$\frac{\sin \frac{N\Delta\omega\Delta t}{2}}{N \sin \frac{\Delta\omega\Delta t}{2}} e^{-j(N-1)\frac{\Delta\omega\Delta t}{2}} \quad (31)$$

The magnitude is plotted in Figure 5.

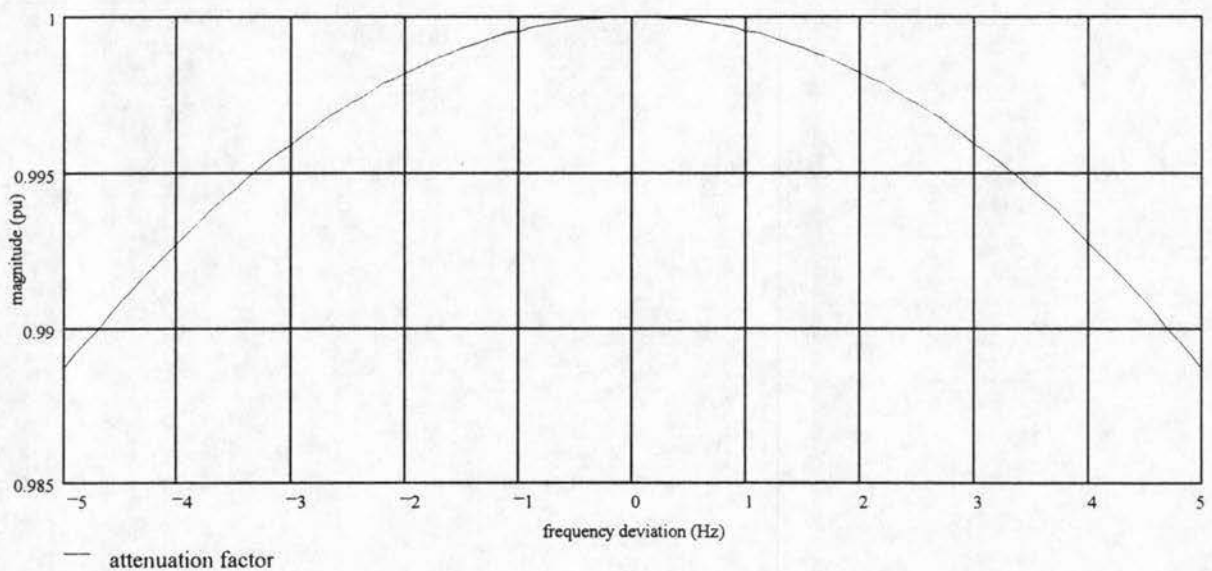


Figure 5. DFT attenuation factor versus frequency.

The phase is plotted in Figure 6.

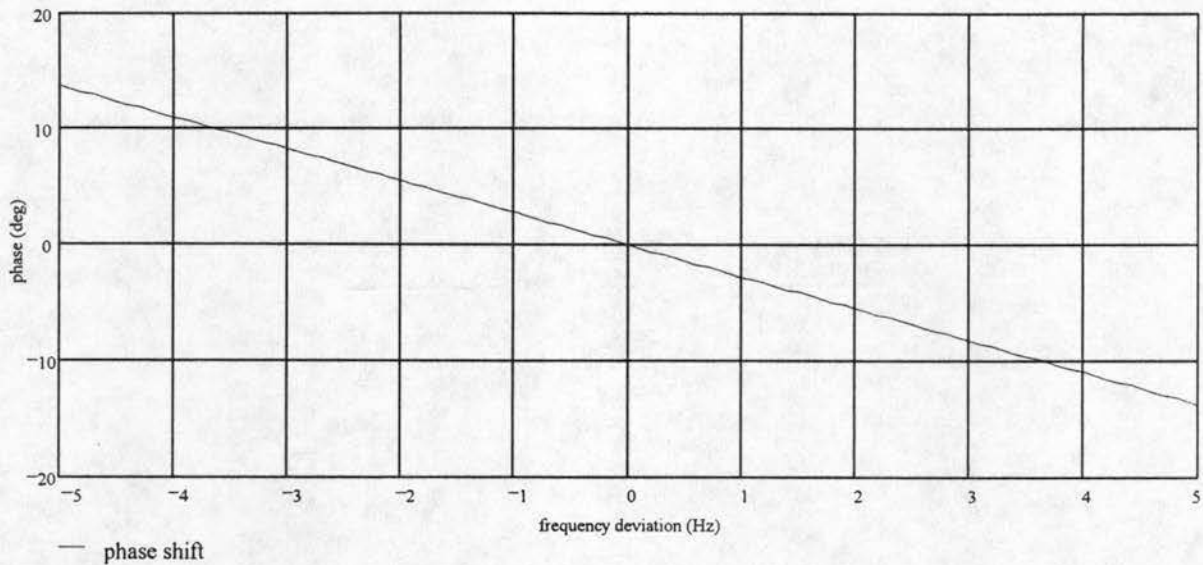


Figure 6. DFT phase shift versus frequency.

During simulations this attenuation and phase shift of the traditional response can be corrected since the frequency is known exactly as it is one of the states.

3.2.2 Aliasing

Aliasing refers to the condition that occurs when frequency components are present at frequencies above the Nyquist frequency or two times the sampling rate. In practice, this is fixed by providing an anti-aliasing filter with a cut-off frequency of $1/3$ to $1/2$ the sampling rate before the waveform is sampled. In simulations, this is not a problem if the step size is small enough.

3.2.3 Picket-fence effect

The preceding graph of an attenuation factor is part of what is known as the picket-fence effect [7]. When the analyzed waveform contains frequencies that are not integer multiples of the fundamental frequency, the energy is spread across several frequencies. The total energy content remains the same.

3.2.4 Leakage

When the DFT window contains only a partial cycle, energy is distributed into adjacent frequencies. When a sliding DFT is used on a waveform with abrupt changes, energy appears at all frequencies due to the leakage effect. Leakage and the picket-fence effect are very similar phenomena. Both occur because the DFT assigns energy to frequencies in discrete steps, sometimes referred to as bins, and because the DFT transformation incurs no loss. If the data being measured contain frequencies that do not fall precisely at integer multiples of the fundamental, energy is leaked into adjacent bins (it has to go somewhere since the transformation incurs no loss). Leakage of spectral energy is caused by applying a window to the signal [8]. A rectangular window is used in the thesis as all samples have equal weight.

4. SIMULATION

4.1 Generator Data

The data for the generator used in the simulation is given in Table 1.

Table 1. Generator data.

$N = 60$	$R_s = 0.0017 pu$	$H = 3.2 \text{ sec.}$	$P = 2$
$w_s = 2\pi 60 \text{ r/s}$	$T'_{do} = 9.6 \text{ sec.}$	$X_d = 2.5 pu$	$X'_d = 0.39 pu$
$X''_d = 0.27 pu$	$X_{ls} = 0.2 pu$	$T''_{do} = 0.08 \text{ sec.}$	$T'_{qo} = 1.7 \text{ sec.}$
$X_q = 2.1 pu$	$X'_q = 0.83 pu$	$X''_q = 0.29 pu$	$T''_{qo} = 0.14 \text{ sec.}$
$T_E = 0.8 \text{ sec.}$	$K_E = 1.0$	$S_E = 0.0$	$T_F = 1.0 \text{ sec.}$
$K_F = 0.06 pu$	$T_A = 0.2 \text{ sec.}$	$K_A = 400 pu$	$V_{R_{min}} = 0 pu$
$V_{R_{max}} = 5.0 pu$	$T_{CH} = 0.3 \text{ sec.}$	$T_{SV} = 0.2 \text{ sec.}$	$R_d = 0.5 pu$
$P_{SV_{min}} = 0 pu$	$P_{SV_{min}} = 5.0 pu$		

The generator is initially running at rated speed w_s and open circuited with $V_t=1.0$ pu. A balanced 0.01 pu resistive load is switched in at time = 0.05 sec. The balanced load resistance is then changed to 2.0 pu at time = 0.15 second. The simulation was performed using fourth-order Runge-Kutta integration with a time step of $1/3600$ second. The positive sequence DFT is calculated using a non-recursive one-cycle algorithm with 60 samples/cycle. The source code is given in Appendix A.

4.2 Full Model Results

Figures 7, 8, and 9 are the per unit a, b, and c currents for the full model.

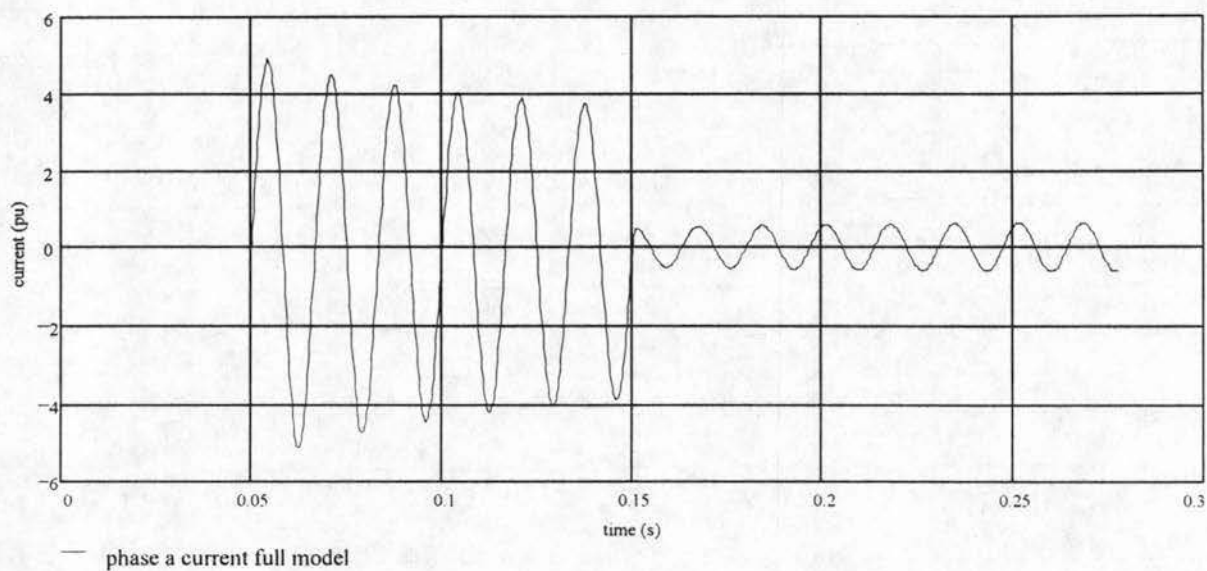


Figure 7. Phase a current - full model.

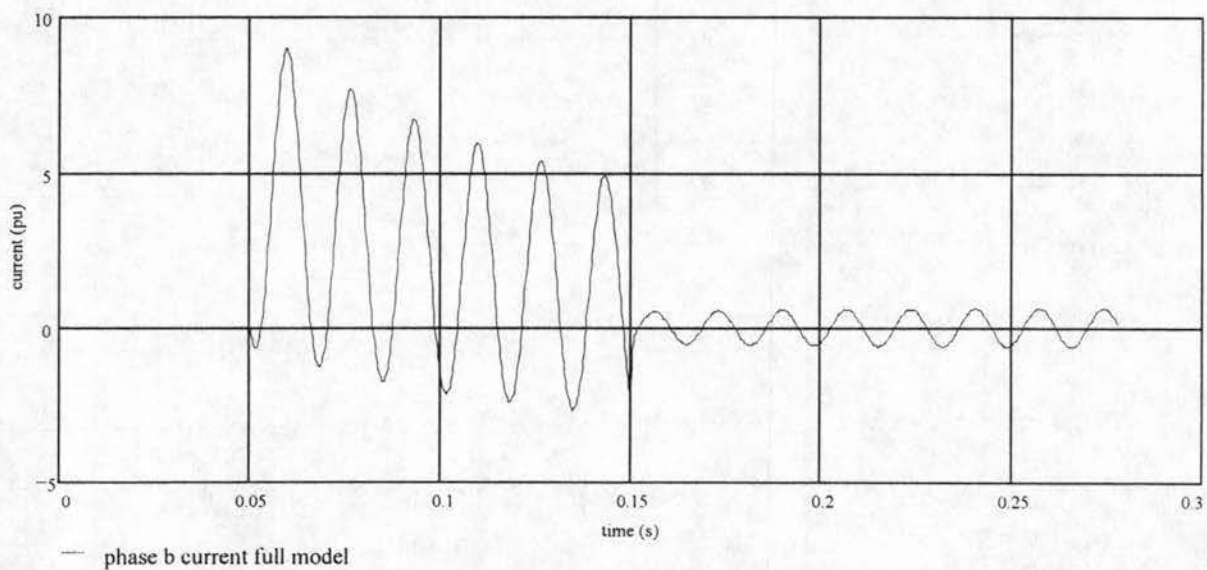


Figure 8. Phase b current - full model.

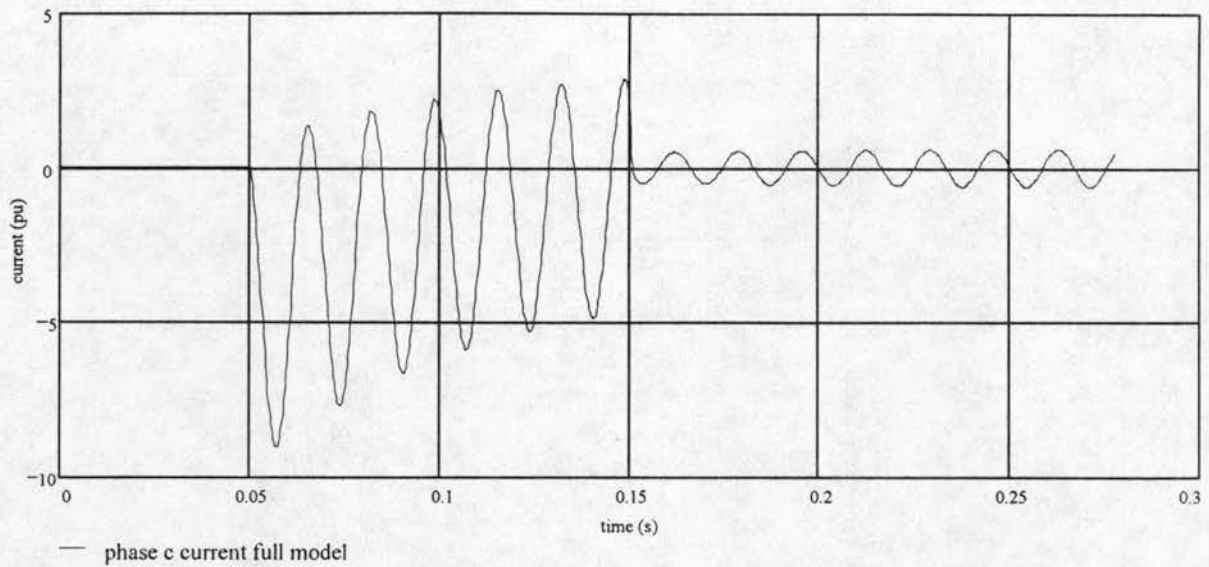


Figure 9. Phase c current - full model.

4.2.1 Full model phasors

Figure 10 gives the exact and measured dynamic phasors for the full model. The magnitude of the measured phasor takes one full cycle to initialize. It acts as a band-pass filter since it measures only the fundamental component.

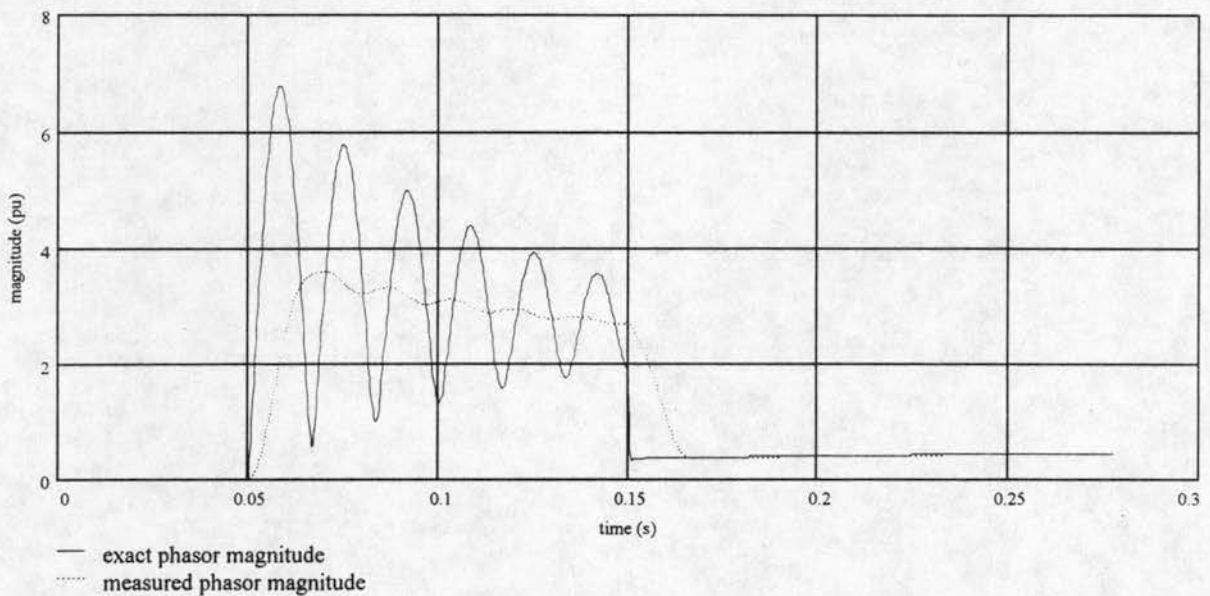


Figure 10. Exact and measured phasor magnitudes - full model.

The phases of the phasors are shown in Figure 11.

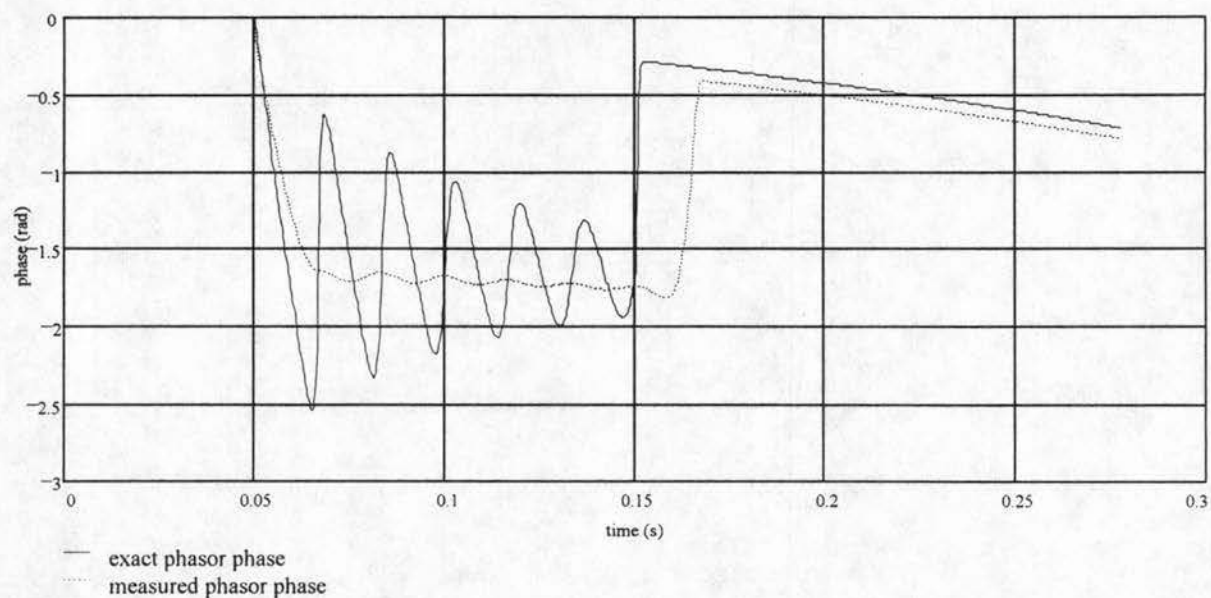


Figure 11. Exact and measured phasor phases - full model.

4.2.2 DFT error due to frequency

To see how much error is in the phasor due to off-nominal frequency, the speed (in Hertz) is plotted in Figure 12.

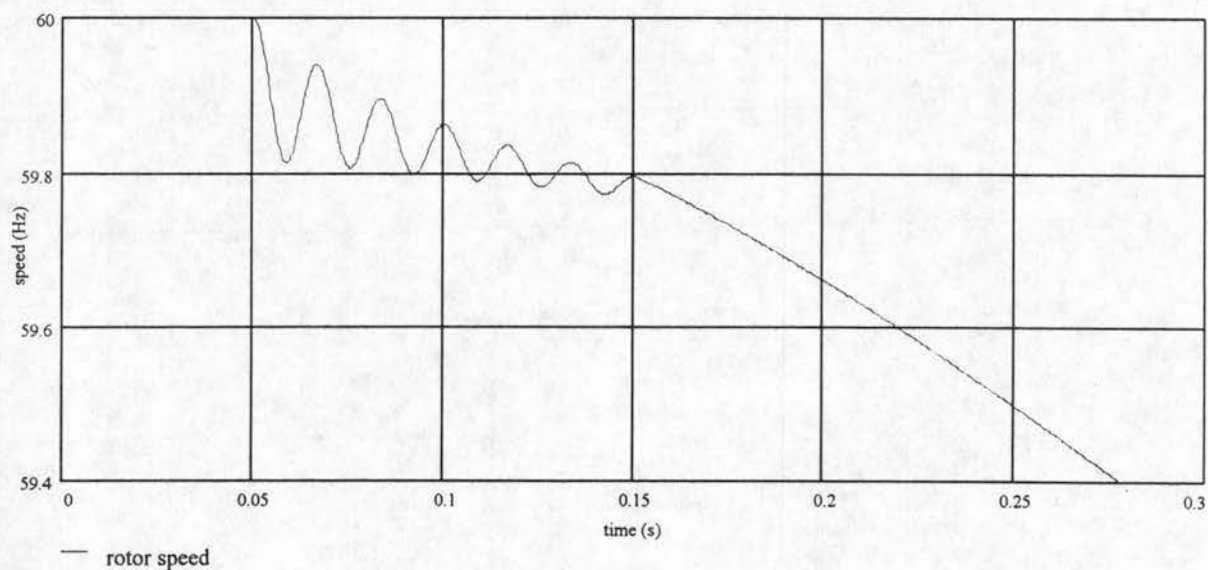


Figure 12. Rotor speed in hertz.

The attenuation error due to frequency (from Figure 5) is less than 0.2 percent.

4.2.3 The dc phasor

It is difficult to tell from the positive sequence DFT of the fundamental that there is a dc component present. The DFT acts as a band-pass filter around the component being measured.

We can perform a DFT based on the dc component by replacing n in Equation (29) with zero.

We can then perform a similar positive sequence DFT based on the dc component by substituting Equation (29) into Equation (30). Figure 13 is the dc phasor of the full model.

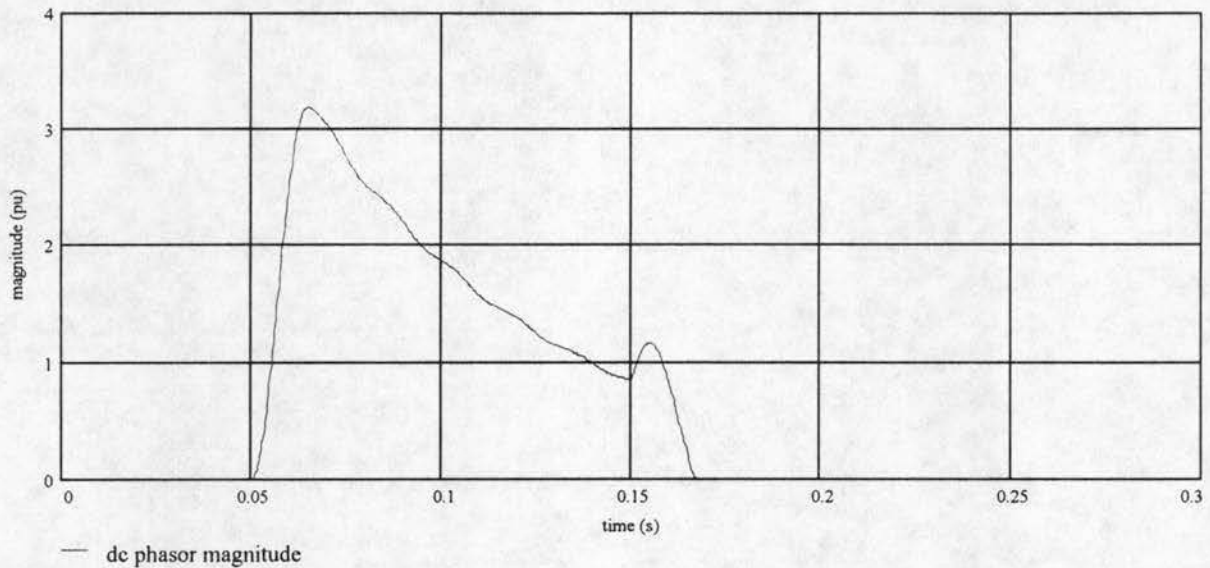


Figure 13. The dc phasor - full model.

The bump in near time = 0.15 is due to the leakage effects of the abrupt change in the waveform.

4.3 Reduced Model Results

The results look very similar to those for the full model without the dc component in the a, b, c currents. Figures 14, 15, and 16 show the a, b, and c currents for the reduced model.

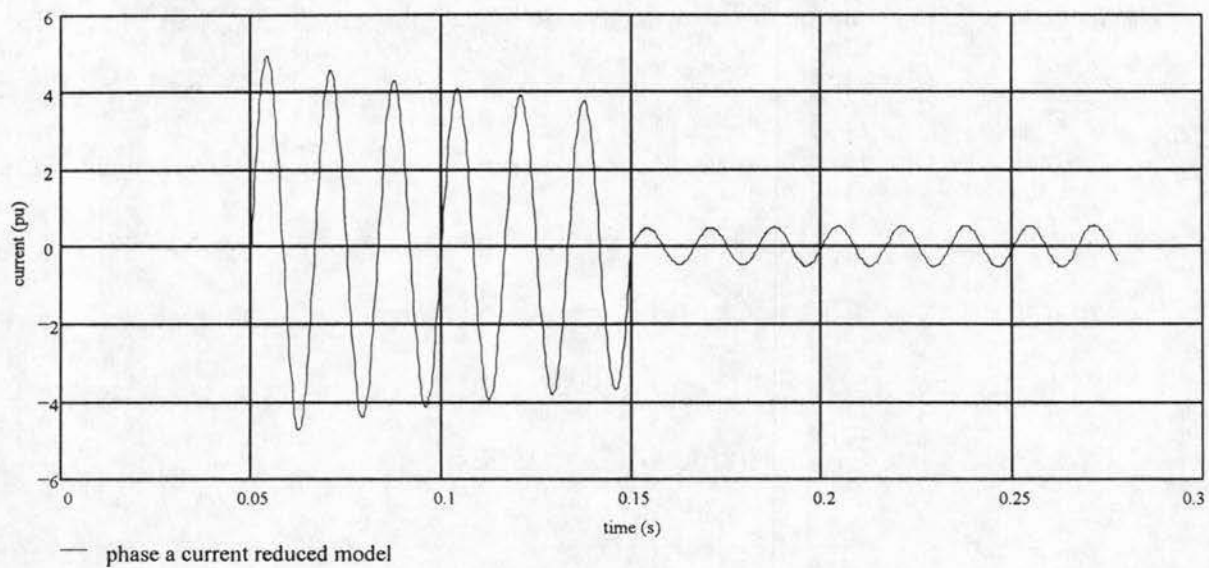


Figure 14. Phase a current - reduced model.

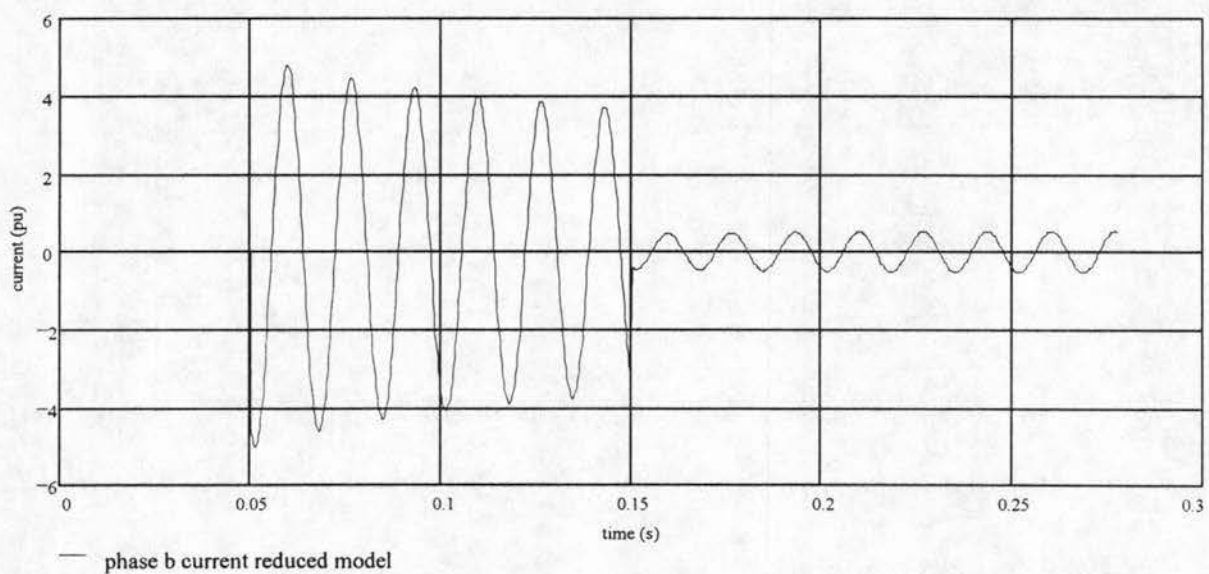


Figure 15. Phase b current - reduced model.

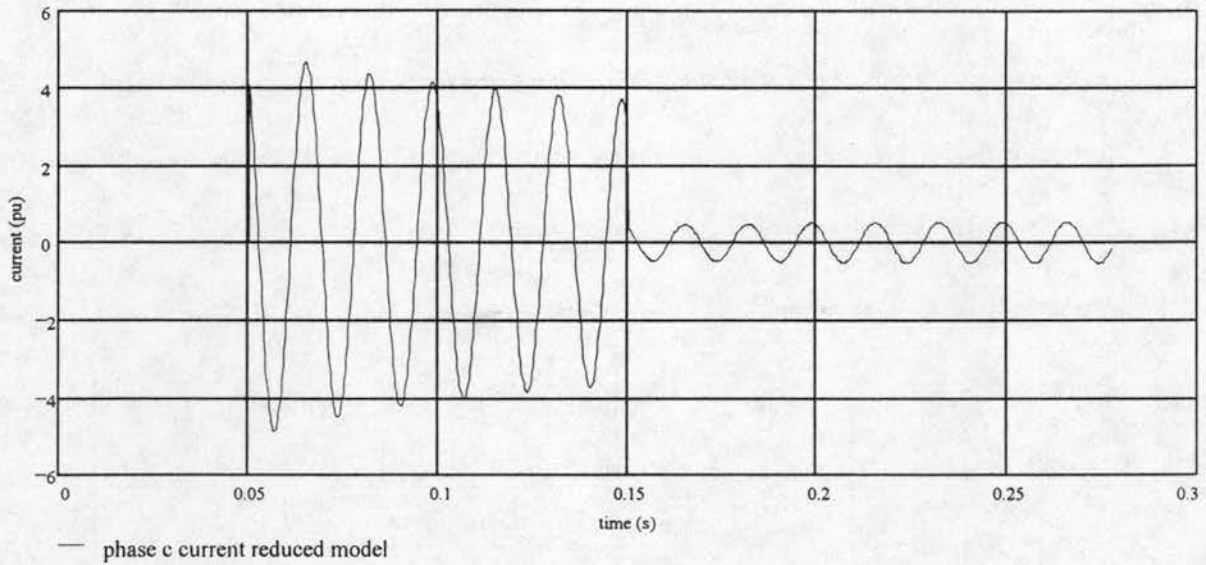


Figure 16. Phase c current - reduced model.

4.3.1 Reduced model phasors

The approximate and approximate measured dynamic phasor magnitudes for the reduced model are shown in Figure 17. Again, the magnitude of the measured phasor takes one full cycle to initialize. It acts as a band-pass filter since it measures only the fundamental component.

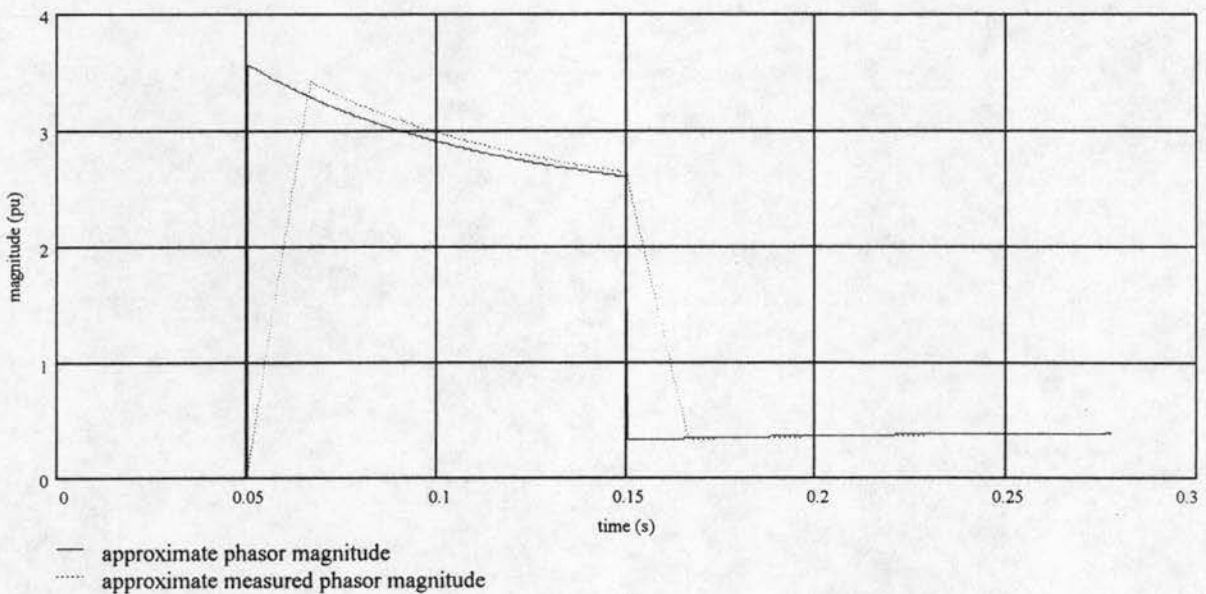


Figure 17. Approximate and approximate measured phasor magnitudes - reduced model.

The measured phasor appears to be very close to the actual phasor. In fact, the measured phasor is a low-pass filtered version of the approximate phasor. The low-pass filtering smooths the abrupt transitions and delays the signal by a half cycle (30 samples). Figure 18 is the approximate phasor and the approximate measured phasor magnitudes shifted forward 30 samples.

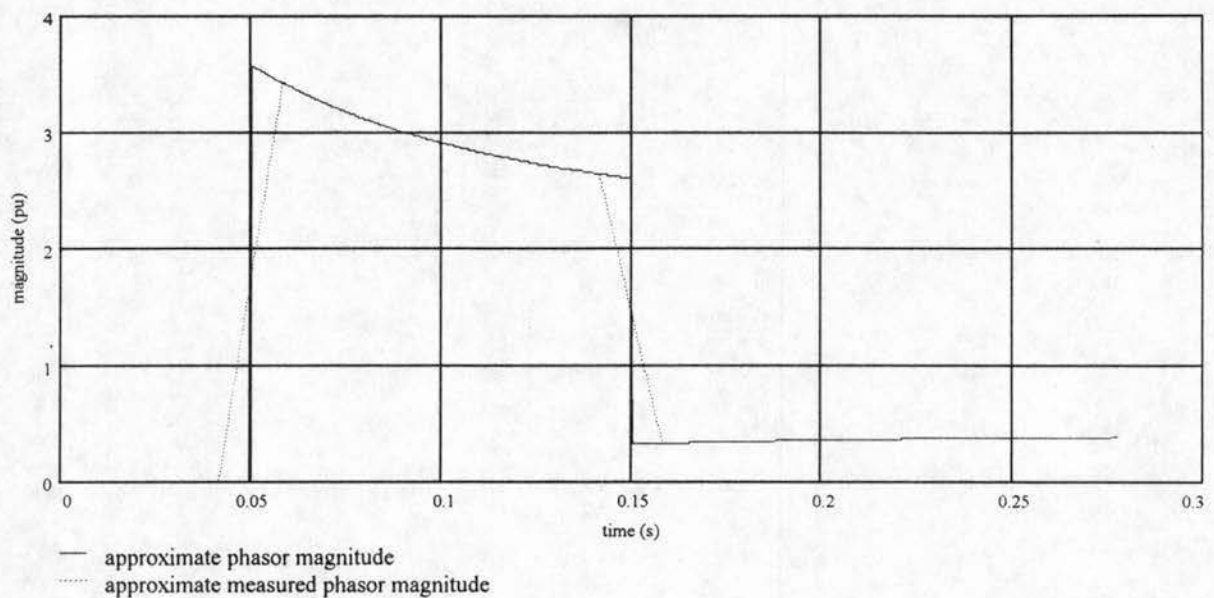


Figure 18. Approximate and approximate measured phasor magnitudes - shifted 30 samples.

Though the approximate measured phasor is not directly useful, it does demonstrate one important fact. With no dc or harmonic components present in the input waveform, the DFT phasor (measured phasor) is a low-pass filtered approximate phasor with no steady-state error. The approximate and approximate measured dynamic phasor phases for the reduced model are shown in Figure 19.

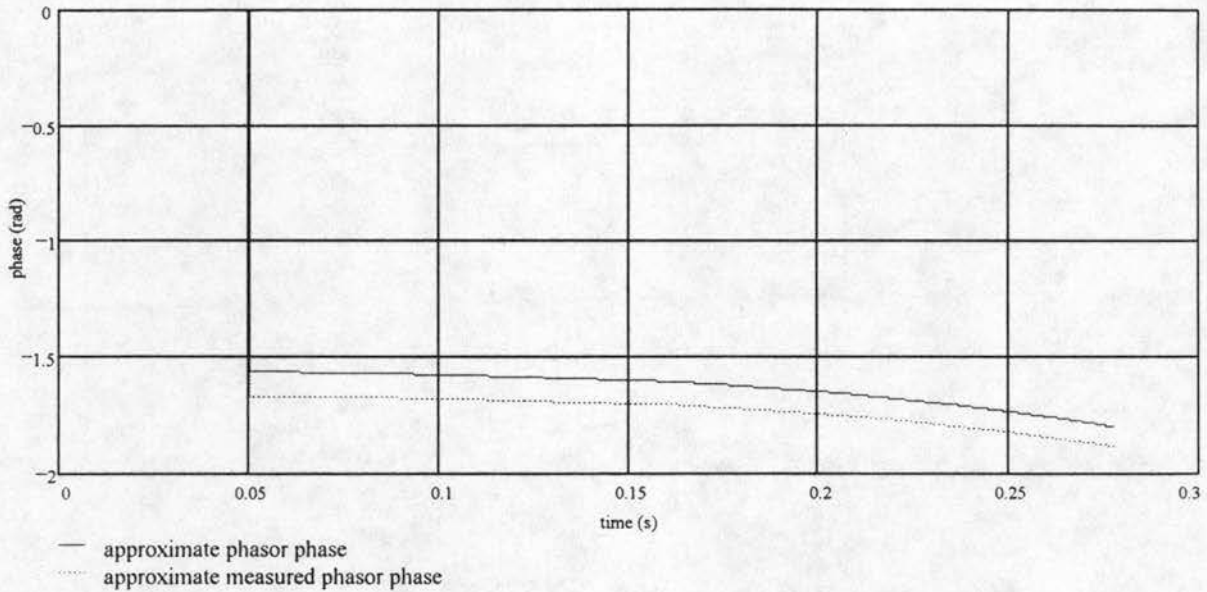


Figure 19. Approximate and approximate measured phasor phases - reduced model.

4.3.2 The dc phasor

It is difficult to tell from the positive sequence DFT of the fundamental if there is a dc component present. The DFT acts as a band-pass filter around the component being measured. We can perform a similar positive sequence DFT based on the dc component by replacing n with zero in Equation (29). For the reduced model, the leakage effects of the dc component positive sequence DFT are more dramatic since there is no dc component in the reduced model abc waveforms.

The dc component is from waveform discontinuities encountered by the DFT. Figure 20 is the dc phasor for the reduced model.

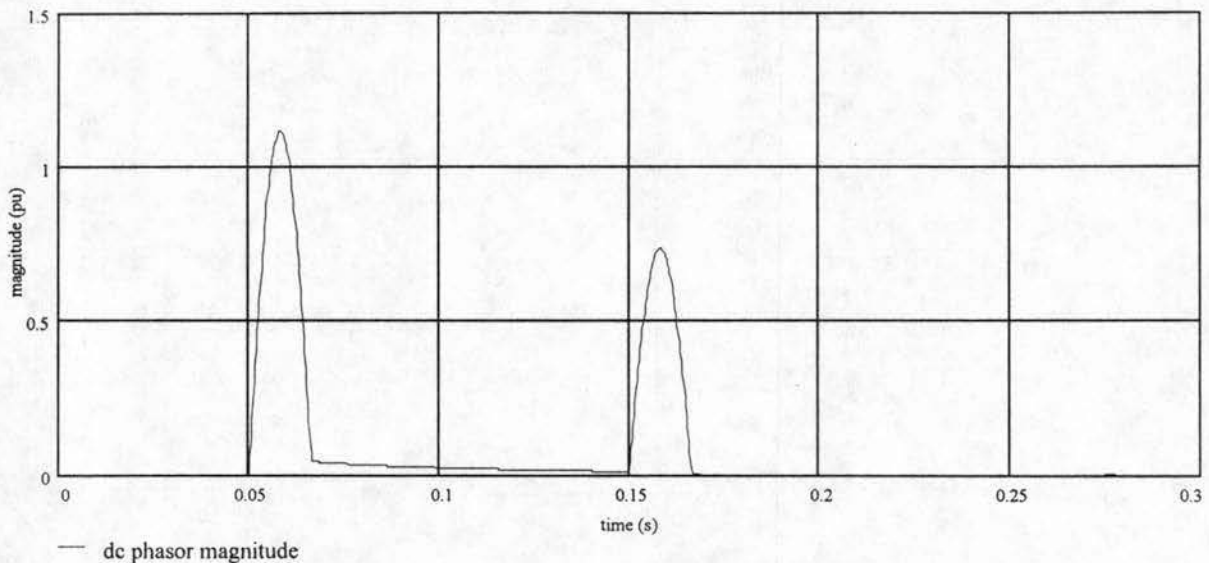


Figure 20. DC phasor - reduced model.

4.4 Half-Cycle DFT Results

It is interesting to investigate the effect of changing the window length of the DFT. In particular, it seems as though reducing the cycle length would reduce the delay of the phasor initialization. Reducing the delay of the phasor initialization would allow the real-time controller to see large changes in the measured quantities sooner, thus providing a better chance for successful control. A controller could use a hybrid approach. The half-cycle DFT could be used for faster response while the full-cycle DFT could be used for better steady-state error. The same simulations were run with a half-cycle DFT. The algorithm for the half-cycle DFT is the same as for the full cycle (Equation (28)) except that N is replaced by $N/2$ and the sum is multiplied by 2. The results from the full model using a half-cycle DFT are shown in Figures 21 and 22.

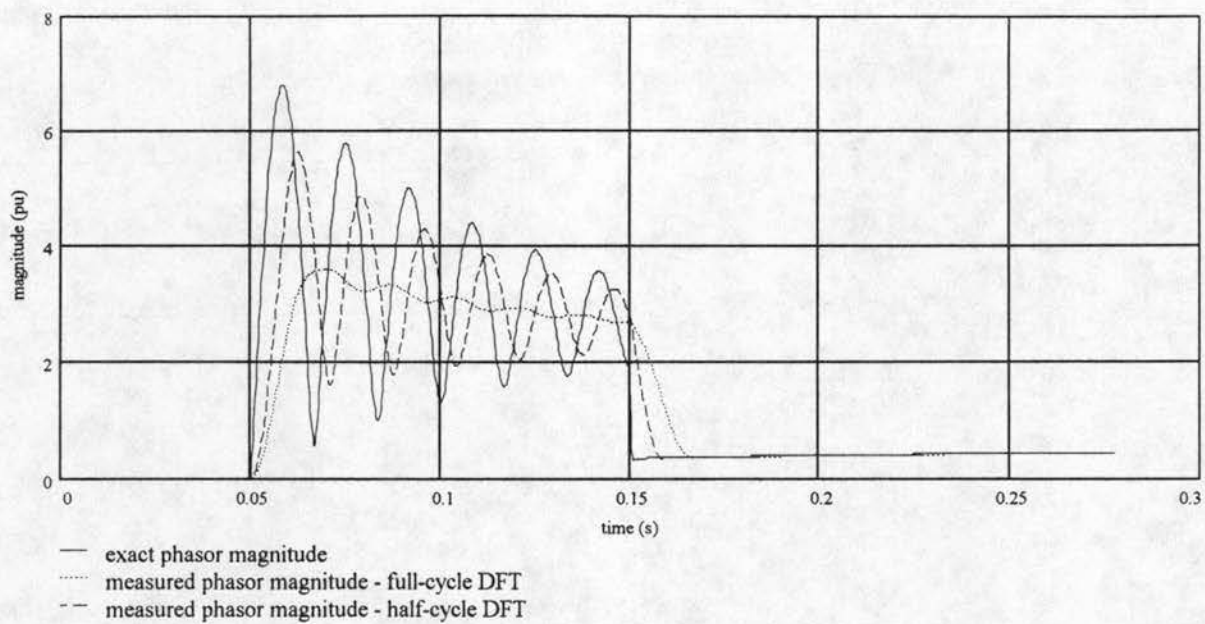


Figure 21. Half-cycle DFT magnitude - full model.

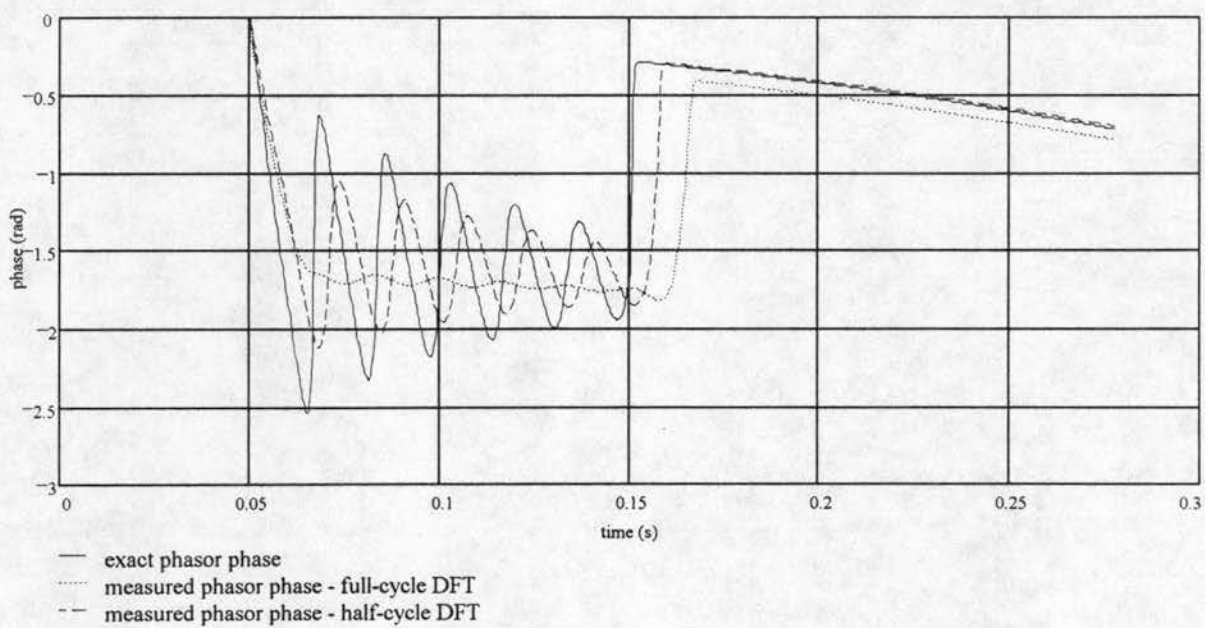


Figure 22. Half-cycle phasor phase - full model.

It appears as though the DFT has lost the ability to filter the dc component. It is informative to look at the frequency response for sine waves with varying angles as in Figures 2, 3, and 4. The frequency response of the half-cycle DFT is shown in Figures 23, 24, and 25.

If the input waveform is $\sin(\omega t + \Theta)$ with ω varied from 0 to 360 Hz and $\Theta = 0$, the frequency response actually has a peak of 1.071 pu versus 1.0 pu for the traditional frequency response.

The frequency response is shown in Figure 23.

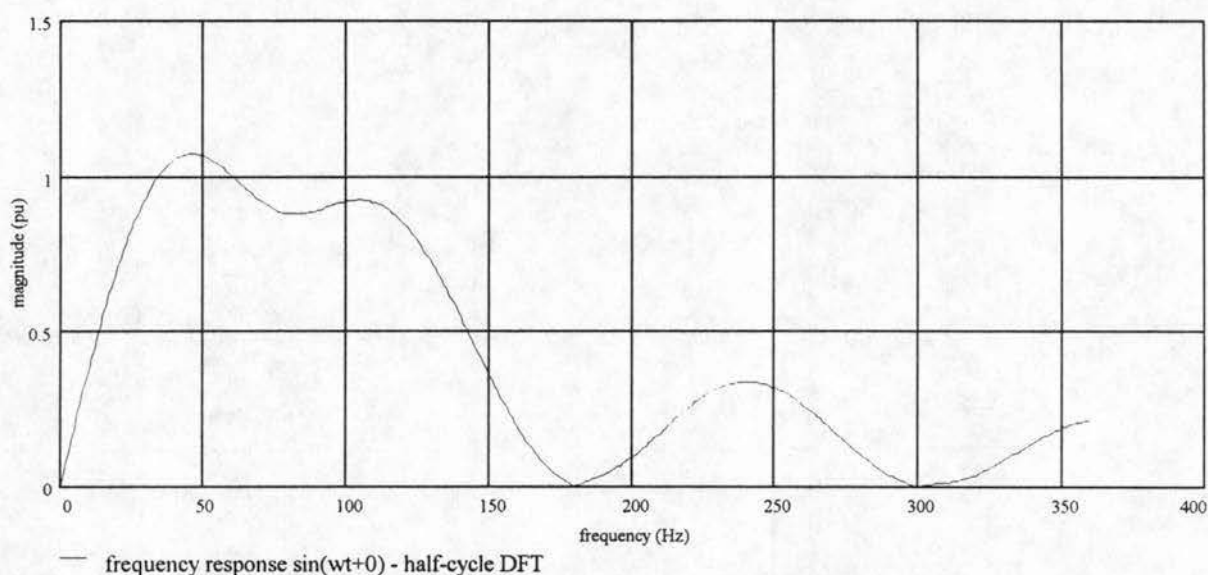


Figure 23. Half-cycle DFT frequency response with input = $\sin(\omega t + 0)$.

If the input waveform is $\sin(\omega t + \Theta)$ with ω varied from 0 to 360 Hz and $\Theta = 45$ degrees, the frequency response is shown in Figure 24.

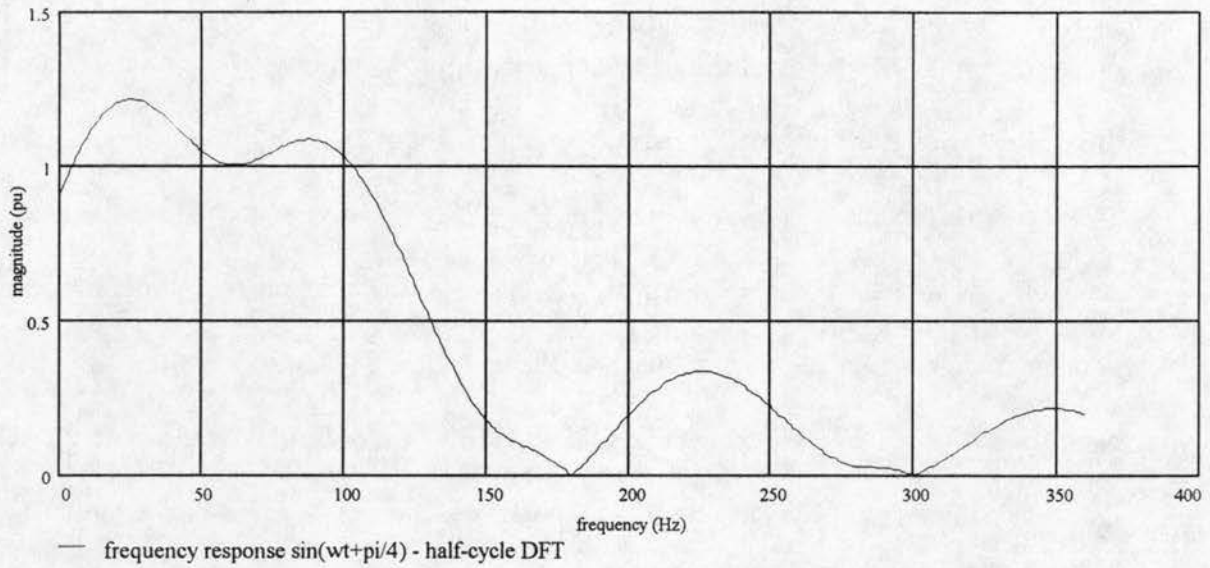


Figure 24. Half-cycle DFT frequency response with input = $\sin(\omega t + \pi/4)$.

If the input waveform is $\sin(\omega t + \Theta)$ with ω varied from 0 to 360 Hz and $\Theta = 90$ degrees, the frequency response is shown in Figure 25.

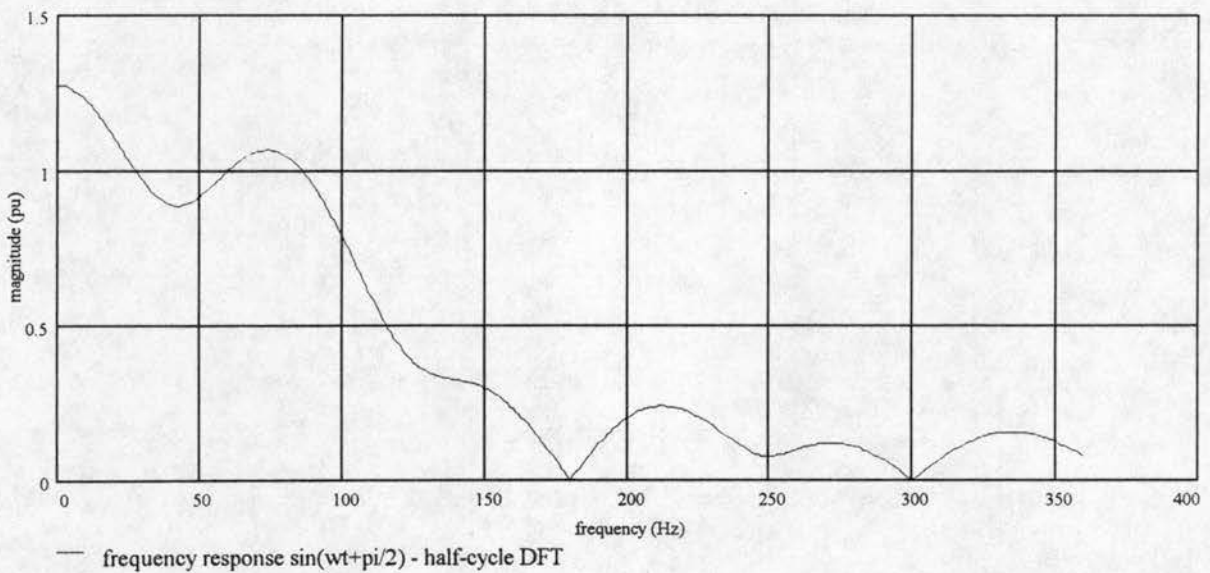


Figure 25. Half-cycle DFT frequency response with input = $\sin(\omega t + \pi/2)$.

In all three graphs, the magnitude of the fundamental component is one. It can be seen from the graphs that if the frequency of the signal being measured is not exactly the fundamental frequency, there is an error. It is obvious that the dc component is not rejected. It is also obvious that the even-order harmonics are also not rejected. Therefore, if there is a second harmonic due to transformer magnetizing inrush, it will not be rejected. Since the reduced-order model results contain no dc component, we would expect the half-cycle DFT to follow the exact phasors more closely. We would also expect the measured phasor to approach the steady-state value more quickly since the DFT takes a half-cycle to initialize. The results of the half-cycle DFT with the reduced-order model are shown in Figures 26 and 27.

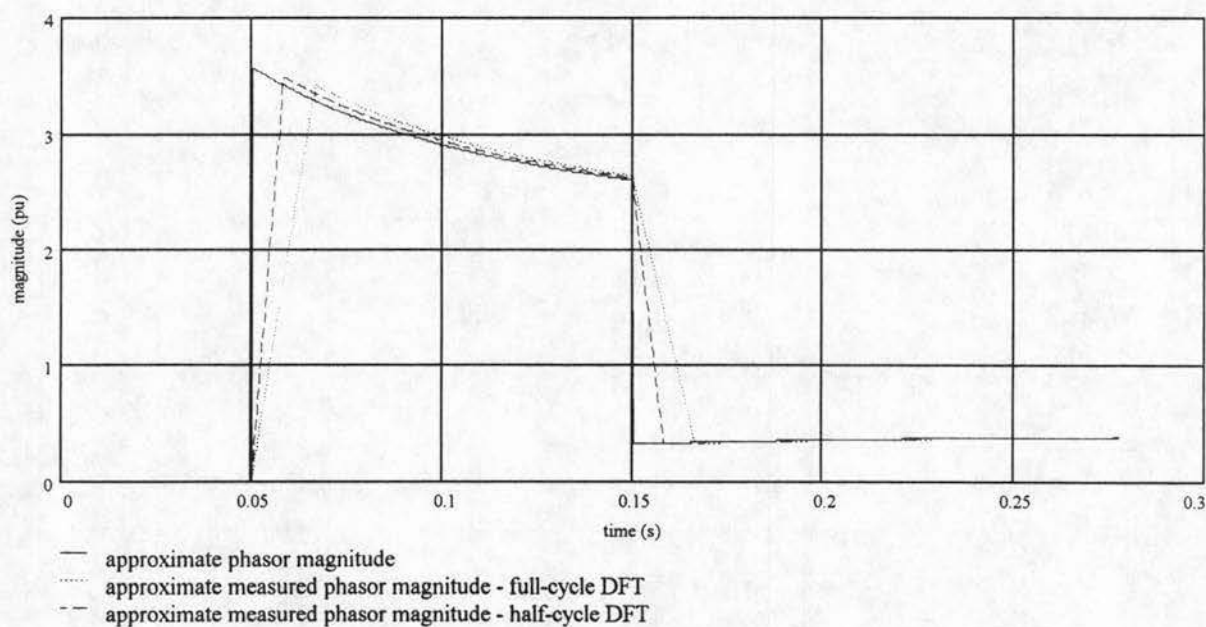


Figure 26. Half-cycle phasor magnitude - reduced model.

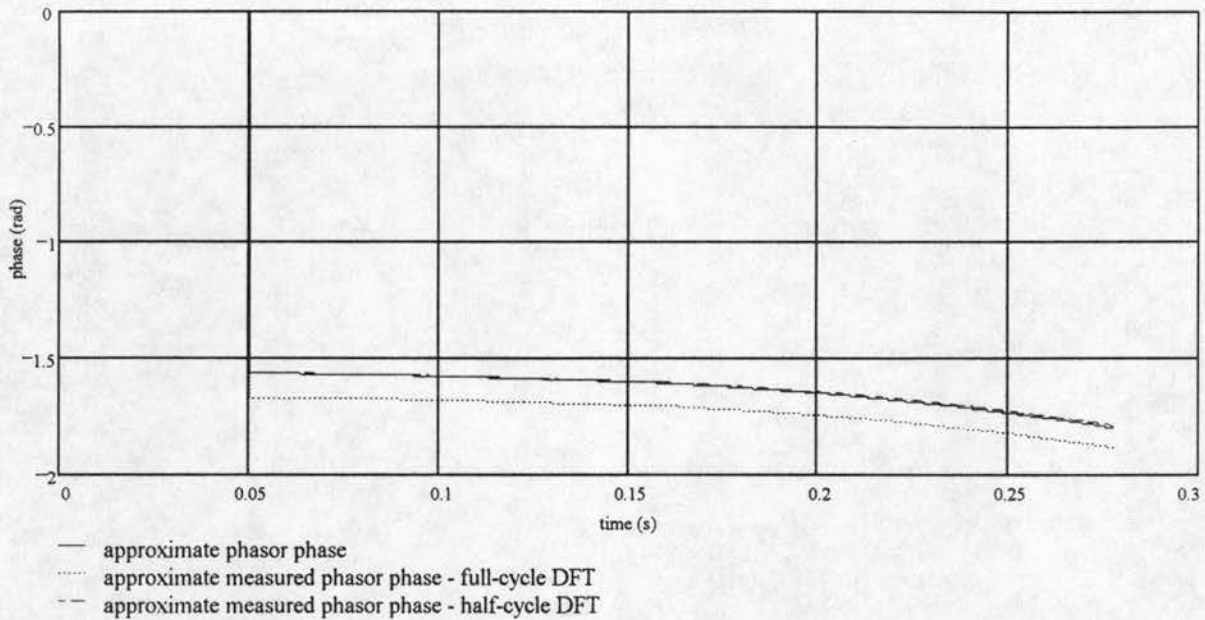


Figure 27. Half-cycle phasor phase - reduced model.

4.5 The dc Component Rejection in One-Cycle DFT

Rejection of the dc component leads us to analyze the full-cycle DFT results of the full model from Figure 10. There is a slight ac component in the measured phasor magnitude, which can be explained by the fact that the abc currents and the exact phasor do not have a dc offset. They have a decaying dc offset. A true dc offset would add no error to a full-cycle DFT. The magnitude of the full-cycle DFT at zero Hertz in Figures 2, 3, and 4 is zero. However, the decaying dc offset has a frequency of about 3 Hz. At 3 Hz the additional magnitude is 10%, 6%, and 1.6%, respectively, from Figures 2, 3, and 4. The percentages are percent of the 3 Hz component. Figure 28 shows the full and reduced model phasors plotted together along with the dc phasor and the absolute value of the difference between the full and reduced model phasors.

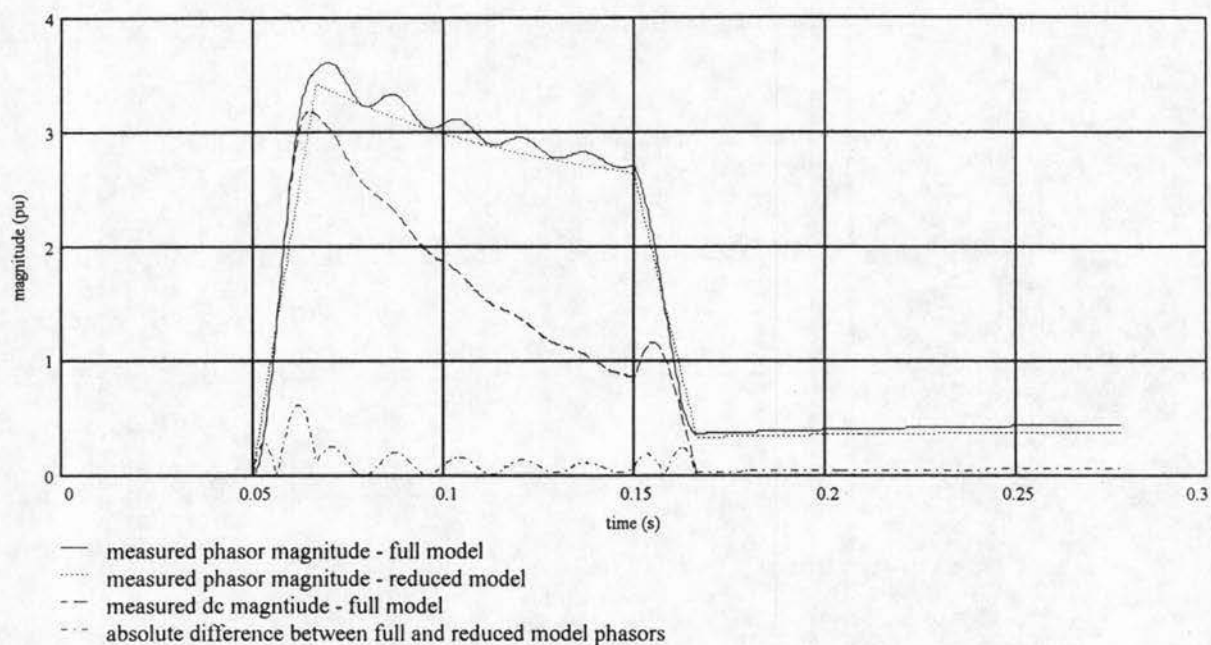


Figure 28. Measured phasor comparison.

To check the estimates of 10%, 6% and 1.6% as the DFT slides, the absolute value of the difference between the full and reduced-order phasors divided by the magnitude of the dc phasor times 100 is shown in Figure 29.

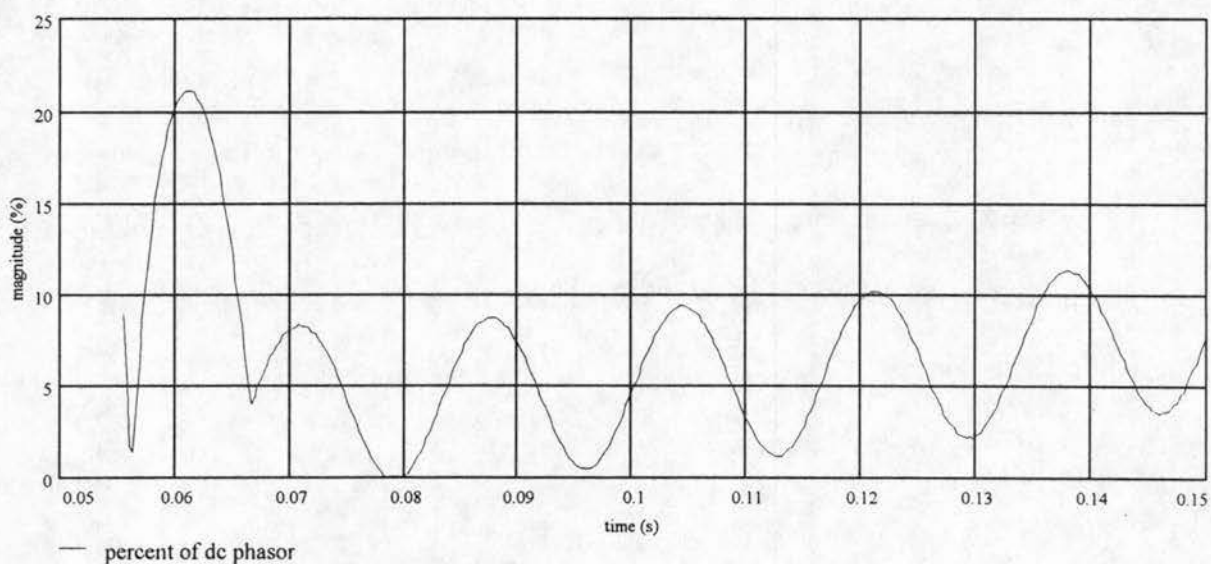


Figure 29. Phasor difference expressed in percent of dc phasor.

The change in magnitude error as the DFT slides causes an ac error component in the phasor measurement. If the frequency response of the DFT were the same as the traditional frequency response as shown in Figure 1, the phasor error would be a fixed 6% at 3 Hz and no ripple would appear in the phasor. After the initial 20% error, the error oscillates between about 10% and 2% as predicted by Figures 2, 3, and 4.

4.6 Phasor Comparison

Figure 30 compares the measured phasor from the full model with the approximate and approximate measured phasors of the reduced model. It can be seen that except for the ripple on the measured phasor, the measured and the approximate measured phasors are very close.

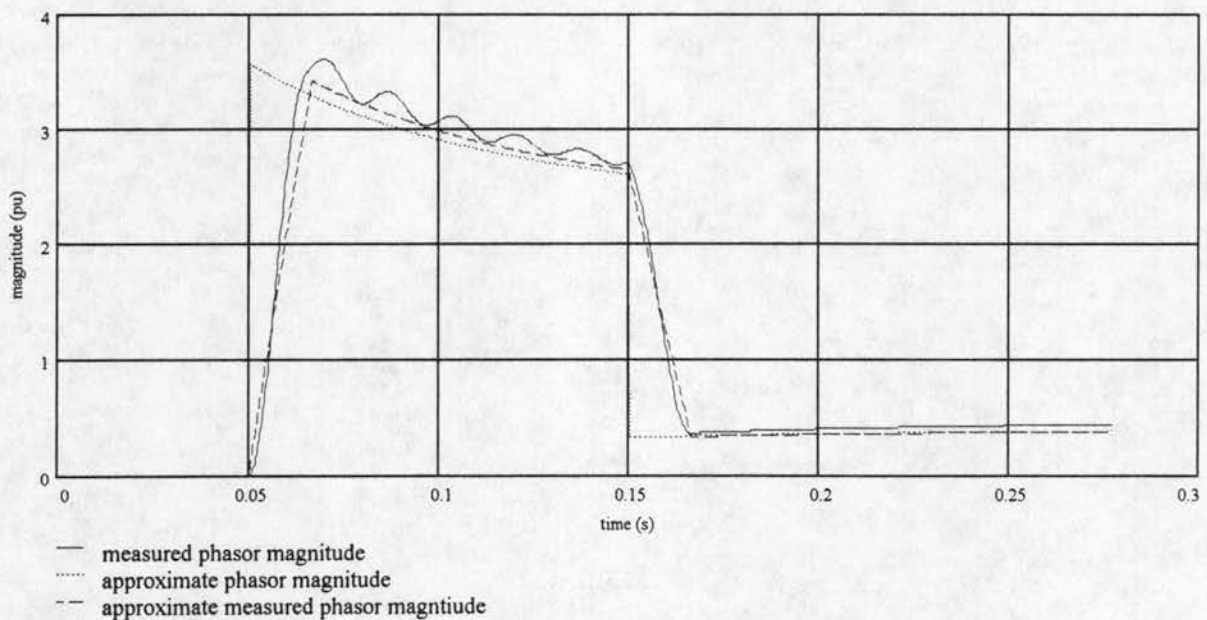


Figure 30. Phasor comparison.

5. CONCLUSIONS

Phasor results of the full model show that the measured phasor is a band-pass filtered version of the exact phasor (Figure 10). The phasor results of the reduced model show that the approximate measured phasor has no steady-state error compared with the approximate phasor (Figure 18) and that the approximate measured phasor is a band-pass filtered (thus delayed) version of the approximate phasor (Figure 17). The measured phasor from the full model is very close to the approximate measured phasor of the reduced model (Figure 30). This implies that the DFT removes the fast stator transients after they occur while the reduced model removes the fast stator transients before they occur. This is significant because real-time control will use measured phasors and/or a reduced-order model. Because both methods yield approximately the same results through totally different means, we can use the reduced-order model with confidence.

REFERENCES

- [1] A. G. Phadke, "Synchronized phasor measurements," *IEEE Computer Applications in Power*, pp. 10-15, April 1993.
- [2] A. G. Phadke and J. S. Thorp, "Improved control and protection of power systems through synchronized phasor measurements," *Control and Dynamic Systems*, vol. 43. New York: Academic Press, 1991, pp. 335-376.
- [3] P. C. Krause, *Analysis of Electric Machinery*. New York: McGraw-Hill, 1986.
- [4] P. W. Sauer, *Power System Dynamics*, 1990 to be published by Prentice-Hall.
- [5] J. L. Blackburn, *Symmetrical Components for Power System Engineering*. New York: Dekker, 1993.
- [6] A. G. Phadke, J. S. Thorp, and M. G. Adamiak, "A new measurement technique for tracking voltage phasors, local system frequency, and rate of change of frequency," *IEEE Transactions on Power Apparatus Systems*, vol. PAS-102, no.5, pp. 1025-1038, May 1983.
- [7] J. Arrillaga, D. A. Bradley, and P. S. Bodger, *Power System Harmonics*. New York, NY: John Wiley and Sons, 1985.
- [8] A. V. Oppenheim and R. W. Schaffer, *Discrete-Time Signal Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1989.

APPENDIX A FORTRAN CODE

The FORTRAN code used for simulations follows. The code is broken up into three sections.

The code used for the full model is given in Appendix A.1. The code for the reduced-order model is given in Appendix A.2. The half-cycle phasor routine is given in Appendix A.3.

A.1 Full Model

```

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CCCCCCCCCCCCCCCC
C
C   POWER SYSTEM MODEL
C   WRITTEN BY: MICHAEL P. RANSICK
C   4TH ORDER RUNGE-KUTTA INTEGRATION
C   POSITIVE SEQUENCE DFT INCLUDED
C   VRMIN/MAX FIXED , PSVMIN/MAX FIXED
C   DATE: 07/10/95
C   FULL MODEL, BALANCED FAULT
C   ONE-CYCLE PHASOR
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CCCCCCCCCCCCCCCC
      DOUBLEPRECISION DER(13),Y(13),P,IA,IB,IC,
1  MAGIDQ,PHASEIDQ,MAGVDQ,PHASEVDQ,WS,RS,VD,VQ,H,TDOPP,TQOP,R,
2  TA,TE,KE,SE,TF,KF,KA,VREF,VT,TCH,TSV,PC,RD,XD,XDP,XDPP,XLS,
3  XQ,XQP,XQPP,CID,CIQ,RLOAD,PI,PSVMIN,PSVMAX,VRMIN,VRMAX,T,
4  DELTAT,DFTMAG,DFTPH

      COMMON /A/WS,RS,VD,VQ,H,TDOP,TDOPP,TQOP,TQOPP,R
1  /B/TA,TE,KE,SE,TF,KF,KA,VREF,VT,TCH,TSV,PC,RD
1  /C/XD,XDP,XDPP,XLS,XQ,XQP,XQPP,CID,CIQ,RLOAD,PI
1  /D/SCALE,PSVMIN,PSVMAX,VRMIN,VRMAX
      DATA DER,Y/26*0.0/

C.....   OPEN FILES TO OUTPUT DATA

C   OPEN(90,FILE='DQ.DAT',STATUS='NEW')
      OPEN(91,FILE='FBIABCDQ.DAT',STATUS='NEW')
      OPEN(92,FILE='FBY4.DAT',STATUS='NEW')
C   OPEN(92,FILE='Y16.DAT',STATUS='NEW')
C   OPEN(93,FILE='Y713.DAT',STATUS='NEW')
      OPEN(94,FILE='FBDFT.DAT',STATUS='NEW')

```

C..... INITIALIZE CONSTANTS

```

VT   = 1
VREF = 1.0+1./400.
TOTALT = 0.5
DELTAT = 1./3600.
T     = 0.0
WS    = 60.*2.*3.14159
N     = 60
XDPP  = 0.27
XQ    = 2.1
TE    = 0.8
KF    = 0.06
VRMAX = 5.0
VRMIN = 0.0
PSVMAX = 5.0
PSVMIN = 0.0
RS    = 0.0017
XLS   = 0.2
XQP   = 0.83
KE    = 1.0
TA    = 0.2
TCH   = 0.3
PC    = 20.0
H     = 3.2
XD    = 2.5
TDOP  = 9.6
TDOPP = 0.08
XQPP  = 0.29
SE    = 0.0
KA    = 400
TSV   = 0.2
P     = 2
XDP   = 0.39
TQOP  = 1.7
TQOPP = 0.14
TF    = 1.0
RD    = 0.05
Y(4)  = WS
PI    = 4.*ATAN(1.0)
SCALE = 0.0
NDP = INT(TOTALT/DELTAT)

```

C..... INITIAL CONDITONS


```

Y(1) = 1.0
Y(2) = 0.0
Y(3) = 0.0
Y(4) = WS
Y(5) = 1.0
Y(6) = 1.0
Y(7) = 0.0
Y(8) = 0.0
Y(9) = 1.0
Y(10) = 0.06
Y(11) = 1.0
Y(12) = 0.0
Y(13) = 0.0
VD = 0.0
VQ = 1.0

```

C..... START MAIN PROGRAM LOOP

```

DO 100 K=0,NDP
  T = K*DELTAT

```

```

    CALL RUNGE(Y,T,DELTAT)
    CALL PHASORS(Y,T,MAGVDQ,PHASEVDQ,MAGIDQ,PHASEIDQ,IA,IB,IC)
    CALL DFT(N,K,IA,IB,IC,DFTMAG,DFTPH)

```

```

    IF (MOD(K,1).EQ.0) THEN

```

```

C      WRITE(90,102) T,CID,CIQ,VD,VQ
      WRITE(91,103) T,IA,IB,IC,MAGIDQ,PHASEIDQ
      WRITE(92,104) T,Y(4)
C      WRITE(92,103) T,Y(1),Y(2),Y(3),Y(4),Y(5),Y(6)
C      WRITE(93,103) T,Y(7),Y(8),Y(9),Y(10),Y(11),Y(12),Y(13)
      WRITE(94,101) T,DFTMAG,DFTPH
101  FORMAT(1X,F11.8,1X,F8.3,1X,F8.3,1X,F8.3,1X,F8.3)
102  FORMAT(1X,F11.8,1X,F8.3,1X,F8.3,1X,F8.3,1X,F8.3,1X,F8.3,1X,F8.3)
103  FORMAT(1X,F11.8,1X,F8.3,1X,F8.3,1X,F8.3,1X,F8.3,1X,F8.3,1X,F8.3,
1    ,1X,F8.3)
104  FORMAT(1X,F11.8,1X,F8.3)
      ENDIF

```

```

100  CONTINUE
    END

```

C.....SUBROUTINE TO PERFORM INTEGRATION BY FOURTH ORDER RUNGE-KUTTA

```

SUBROUTINE RUNGE(Y,T,DELTAT)
  DOUBLEPRECISION Y(13),DER(13),K1(13),K2(13),K3(13),K4(13),
1 YTEMP(13),T,DELTAT

```

```

  CALL DERIVE(Y,T,K1)
  DO 10 I=1,13
    K1(I)=DELTAT*K1(I)
    YTEMP(I)=Y(I)+K1(I)/2.
    CALL CURRENT(YTEMP,T)
    CALL LIMITS(YTEMP,T)
10  CONTINUE
    CALL DERIVE(YTEMP,T+DELTAT/2.,K2)
    DO 20 I=1,13
      K2(I)=DELTAT*K2(I)
      YTEMP(I)=Y(I)+K2(I)/2.
      CALL CURRENT(YTEMP,T)
      CALL LIMITS(YTEMP,T)
20  CONTINUE
    CALL DERIVE(YTEMP,T+DELTAT/2.,K3)
    DO 30 I=1,13
      K3(I)=DELTAT*K3(I)
      YTEMP(I)=Y(I)+K3(I)
      CALL CURRENT(YTEMP,T)
      CALL LIMITS(YTEMP,T)
30  CONTINUE
    CALL DERIVE(YTEMP,T+DELTAT,K4)
    DO 40 I=1,13
      K4(I)=DELTAT*K4(I)
      Y(I)=Y(I)+(1./6.)*(K1(I)+2.*K2(I)+2.*K3(I)+K4(I))
      CALL CURRENT(Y,T)
      CALL LIMITS(Y,T)
40  CONTINUE
  RETURN
  END

```

C..... THE STATES

C.....	Y(1)=PD	Y(6)=P1D	Y(11)=VR
C.....	Y(2)=PQ	Y(7)=EDP	Y(12)=TM
C.....	Y(3)=D	Y(8)=P2Q	Y(13)=PSV
C.....	Y(4)=W	Y(9)=EFD	
C.....	Y(5)=EQP	Y(10)=RF	

C..... SUBROUTINE TO CALCULATE THE DERIVATIVES OF THE STATES

```

SUBROUTINE DERIVE(Y,T,DER)
  DOUBLEPRECISION DER(13),Y(13),N,P,IA,IB,IC,
1  MAGIDQ,PHASEIDQ,MAGVDQ,PHASEVDQ,WS,RS,VD,VQ,H,TDOPP,TQOP,R,
2  TA,TE,KE,SE,TF,KF,KA,VREF,VT,TCH,TSV,PC,RD,XD,XDP,XDPP,XLS,
3  XQ,XQP,XQPP,CID,CIQ,RLOAD,PI,PSVMIN,PSVMAX,VRMIN,VRMAX,T,
4  T1,T2,X1,X2,X3,X4

  COMMON /A/WS,RS,VD,VQ,H,TDOP,TDOPP,TQOP,TQOPP,R
1  /B/TA,TE,KE,SE,TF,KF,KA,VREF,VT,TCH,TSV,PC,RD
1  /C/XD,XDP,XDPP,XLS,XQ,XQP,XQPP,CID,CIQ,RLOAD,PI

  T1  = (XDP-XDPP)/((XDP-XLS)**2)
  T2  = (XQP-XQPP)/((XQP-XLS)**2)
  X1  = XD-XDP
  X2  = XDP-XLS
  X3  = XQ-XQP
  X4  = XQP-XLS
  DER(1) = WS*(RS*CID+Y(4)*(Y(2)/WS)+VD)
  DER(2) = WS*(RS*CIQ-Y(4)*(Y(1)/WS)+VQ)
  DER(3) = Y(4)-WS
  DER(4) = (WS/(2.*H))*(Y(12)-Y(1)*CIQ+Y(2)*CID)
  DER(5) = (-Y(5)-X1*(CID-T1*(Y(6)+X2*CID-Y(5)))+Y(9))/TDOP
  DER(6) = (-Y(6)+Y(5)-(X2)*CID)/TDOPP
  DER(7) = (-Y(7)+X3*(CIQ-T2*(Y(8)+X4*CIQ+Y(7)))/TQOP
  DER(8) = (1./TQOPP)*(-Y(8)-Y(7)-(X4)*CIQ)
  DER(9) = (1./TE)*(-(KE+SE)*Y(9)+Y(11))
  DER(10) = (1./TF)*(-Y(10)+(KF/TF)*Y(9))
  DER(11) = (1./TA)*(-Y(11)+KA*Y(10)-(KA*KF/TF)*Y(9)+KA*(VREF-VT))
  DER(12) = (1./TCH)*(-Y(12)+Y(13))
  DER(13) = (1./TSV)*(-Y(13)+PC-(1./RD)*(Y(4)/WS))
  RETURN
  END

```

C..... SUBROUTINE TO CALCULATE CURRENT, VOLTAGE AND SET LOAD

```

SUBROUTINE CURRENT(Y,T)

  DOUBLEPRECISION DER(13),Y(13),N,P,IA,IB,IC,
1  MAGIDQ,PHASEIDQ,MAGVDQ,PHASEVDQ,WS,RS,VD,VQ,H,TDOPP,TQOP,R,
2  TA,TE,KE,SE,TF,KF,KA,VREF,VT,TCH,TSV,PC,RD,XD,XDP,XDPP,XLS,
3  XQ,XQP,XQPP,CID,CIQ,RLOAD,PI,PSVMIN,PSVMAX,VRMIN,VRMAX,T,
4  TEMP1D,TEMP2D,TEMP1Q,TEMP2Q

  COMMON /A/WS,RS,VD,VQ,H,TDOP,TDOPP,TQOP,TQOPP,R
1  /B/TA,TE,KE,SE,TF,KF,KA,VREF,VT,TCH,TSV,PC,RD

```


1 /C/XD,XDP,XDPP,XLS,XQ,XQP,XQPP,CID,CIQ,RLOAD,PI

```

TEMP1D = (XDPP-XLS)/(XDP-XLS)
TEMP2D = (XDP-XDPP)/(XDP-XLS)
TEMP1Q = (XQPP-XLS)/(XQP-XLS)
TEMP2Q = (XQP-XQPP)/(XQP-XLS)

```

IF (T.LT.0.05) THEN

CID = 0.0

CIQ = 0.0

ENDIF

IF ((T.GE.0.05).AND.(T.LT.0.15)) THEN

RLOAD=0.01

CID = (1./XDPP)*(-Y(1)+TEMP1D*Y(5)+TEMP2D*Y(6))

CIQ = (1./XQPP)*(-Y(2)-TEMP1Q*Y(7)+TEMP2Q*Y(8))

VD = CID*RLOAD

VQ = CIQ*RLOAD

VT = SQRT((VD**2)+(VQ**2))

ENDIF

IF (T.GE.0.15) THEN

RLOAD=2.0

CID = (1./XDPP)*(-Y(1)+TEMP1D*Y(5)+TEMP2D*Y(6))

CIQ = (1./XQPP)*(-Y(2)-TEMP1Q*Y(7)+TEMP2Q*Y(8))

VD = CID*RLOAD

VQ = CIQ*RLOAD

VT = SQRT((VD**2)+(VQ**2))

ENDIF

RETURN

END

C..... SUBROUTINE TO CALCULATE PHASORS AND ABC CURRENTS

SUBROUTINE

PHASORS(Y,T,MAGVDQ,PHASEVDQ,MAGIDQ,PHASEIDQ,IA,IB,IC)

DOUBLEPRECISION DER(13),Y(13),N,P,IA,IB,IC,

1 MAGIDQ,PHASEIDQ,MAGVDQ,PHASEVDQ,WS,RS,VD,VQ,H,TDOPP,TQOP,R,

2 TA,TE,KE,SE,TF,KF,KA,VREF,VT,TCH,TSV,PC,RD,XD,XDP,XDPP,XLS,

3 XQ,XQP,XQPP,CID,CIQ,RLOAD,PI,PSVMIN,PSVMAX,VRMIN,VRMAX,T,

4 TEMPVD,TEMPID

COMMON /A/WS,RS,VD,VQ,H,TDOP,TDOPP,TQOP,TQOPP,R

1 /B/TA,TE,KE,SE,TF,KF,KA,VREF,VT,TCH,TSV,PC,RD

1 /C/XD,XDP,XDPP,XLS,XQ,XQP,XQPP,CID,CIQ,RLOAD,PI

1 /D/SCALE,PSVMIN,PSVMAX,VRMIN,VRMAX


```

MAGVDQ = VT
IF (ABS(VD).LT.0.00001) THEN
    TEMPVD=0.00001
ELSE
    TEMPVD=VD
ENDIF
PHASEVDQ=(ATAN2(VQ,TEMPVD))+Y(3)-PI/2.
MAGIDQ=SQRT((CID**2)+(CIQ**2))

TEMPID=CID
IF (ABS(TEMPID).LT.0.00001) THEN
    TEMPID=0.00001
ELSE
    TEMPID=CID
ENDIF
PHASEIDQ=(ATAN2(CIQ,TEMPID))+Y(3)-(PI/2.)+(SCALE*2.*PI)
IF (PHASEIDQ.LT.-PI) THEN
    SCALE=SCALE+1.0
ELSEIF (PHASEIDQ.GT.PI) THEN
    SCALE=SCALE-1.0
ENDIF

PHASEIDQ=(ATAN2(CIQ,TEMPID))+Y(3)-(PI/2.)+(SCALE*2.*PI)

IF (T.LT.0.05) THEN
    PHASEIDQ=0.0
ENDIF

IA=SQRT(2.)*MAGIDQ*COS(WS*T+PHASEIDQ)
IB=SQRT(2.)*MAGIDQ*COS(WS*T+PHASEIDQ-2.*PI/3.)
IC=SQRT(2.)*MAGIDQ*COS(WS*T+PHASEIDQ+2.*PI/3.)

RETURN
END

```

C..... SUBROUTINE TO CALCULATE DFT'S

```

SUBROUTINE DFT(N,K,IA,IB,IC,DFTMAG,DFTPH)
DOUBLEPRECISION DFTMAG,DFTPH,IA,IB,IC,PI
COMPLEX PHIA,PHIB,PHIC,ALPHA,X1,SPHIA(60),SPHIB(60),SPHIC(60),
1 SPHIA2(60),SPHIB2(60),SPHIC2(60)

PI=4.*ATAN(1.0)

```

```

KCY=KCY+1
ALPHA=CMPLX(COS(2.*PI/3.),SIN(2.*PI/3.))

RIA=(SQRT(2.)/N)*IA*COS(2.*KCY*PI/N)
XIA=(SQRT(2.)/N)*IA*(-SIN(2.*KCY*PI/N))
SPHIA2(KCY)=CMPLX(RIA,XIA)
PHIA=PHIA+CMPLX(RIA,XIA)
APHIA=CABS(PHIA)
RIA=REAL(PHIA)
XIA=AIMAG(PHIA)
TEMPRIA=RIA
IF (ABS(TEMPRIA).LT.0.00001) THEN
    TEMPRIA=0.00001
ELSE
    TEMPRIA=RIA
ENDIF
PPHIA=ATAN2(XIA,TEMPRIA)

RIB=(SQRT(2.)/N)*IB*COS(2.*KCY*PI/N)
XIB=(SQRT(2.)/N)*IB*(-SIN(2.*KCY*PI/N))
SPHIB2(KCY)=CMPLX(RIB,XIB)
PHIB=PHIB+CMPLX(RIB,XIB)
APHIB=CABS(PHIB)
RIB=REAL(PHIB)
XIB=AIMAG(PHIB)
TEMPRIB=RIB
IF (ABS(TEMPRIB).LT.0.00001) THEN
    TEMPRIB=0.00001
ELSE
    TEMPRIB=RIB
ENDIF
PPHIB=ATAN2(XIB,TEMPRIB)

RIC=(SQRT(2.)/N)*IC*COS(2.*KCY*PI/N)
XIC=(SQRT(2.)/N)*IC*(-SIN(2.*KCY*PI/N))
SPHIC2(KCY)=CMPLX(RIC,XIC)
PHIC=PHIC+CMPLX(RIC,XIC)
APHIC=CABS(PHIC)
RIC=REAL(PHIC)
XIC=AIMAG(PHIC)
TEMPRIC=RIC
IF (ABS(TEMPRIC).LT.0.00001) THEN
    TEMPRIC=0.00001
ELSE
    TEMPRIC=RIC

```

```

ENDIF
PPHIC=ATAN2(XIC,TEMPRIC)

X1=(1./3.)*(PHIA+ALPHA*PHIB+ALPHA*ALPHA*PHIC)
C NEG  X1=(1./3.)*(PHIA+ALPHA*ALPHA*PHIB+ALPHA*PHIC)
C ZERO  X1=(1./3.)*(PHIA+PHIB+PHIC)
        AX1=CABS(X1)
        TEMPX1=REAL(X1)
        IF (ABS(TEMPX1).LT.0.00001) THEN
            TEMPX1=0.00001
        ELSE
            TEMPX1=REAL(X1)
        ENDIF
        PX1=ATAN2(AIMAG(X1),TEMPX1)

DFTMAG=AX1
DFTPH=PX1

IF (KCY.EQ.N) THEN
    DO 400 I=1,N
        SPHIA(I)=SPHIA2(I)
        SPHIB(I)=SPHIB2(I)
        SPHIC(I)=SPHIC2(I)
        KCY=0
400    CONTINUE
ENDIF

IF (K.GE.N) THEN
    PHIA=PHIA-SPHIA(1)
    PHIB=PHIB-SPHIB(1)
    PHIC=PHIC-SPHIC(1)
    DO 401 J=2,N
        SPHIA(J-1)=SPHIA(J)
        SPHIB(J-1)=SPHIB(J)
        SPHIC(J-1)=SPHIC(J)
401    CONTINUE
ENDIF

RETURN
END

SUBROUTINE LIMITS(Y,T)

DOUBLEPRECISION Y(13),T,VRMIN,VRMAX,PSVMIN,PSVMAX

```

```
COMMON /D/SCALE,PSVMIN,PSVMAX,VRMIN,VRMAX
```

```
      IF (Y(11).LT.VRMIN) THEN
        Y(11)=VRMIN
      ENDIF
      IF (Y(11).GT.VRMAX) THEN
        Y(11)=VRMAX
      ENDIF
      IF (Y(13).LT.PSVMIN) THEN
        Y(13)=PSVMIN
      ENDIF
      IF (Y(13).GT.PSVMAX) THEN
        Y(13)=PSVMAX
      ENDIF
    RETURN
  END
```

A.2 Reduced Model

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CCCCCCCCCCCCCCCC
```

```
C
C   POWER SYSTEM MODEL
C   WRITTEN BY: MICHAEL P. RANSICK
C   4TH ORDER RUNGE-KUTTA INTEGRATION
C   REDUCED ORDER MODEL
C   POSITIVE SEQUENCE DFT INCLUDED
C   VRMIN/MAX,PSVMIN/MAX FIXED
C   DATE: 07/10/95
C   BALANCED FAULT
C   ONE-CYCLE PHASOR
C
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CCCCCCCCCCCCCCCC
```

```
      DOUBLEPRECISION DER(13),Y(13),P,IA,IB,IC,
1  MAGIDQ,PHASEIDQ,MAGVDQ,PHASEVDQ,WS,RS,VD,VQ,H,TDOPP,TQOP,R,
2  TA,TE,KE,SE,TF,KF,KA,VREF,VT,TCH,TSV,PC,RD,XD,XDP,XDPP,XLS,
3  XQ,XQP,XQPP,CID,CIQ,RLOAD,PI,PSVMIN,PSVMAX,VRMIN,VRMAX,T,
4  DELTAT,DFTMAG,DFTPH
```

```
      COMMON /A/WS,RS,VD,VQ,H,TDOP,TDOPP,TQOP,TQOPP,R
1  /B/TA,TE,KE,SE,TF,KF,KA,VREF,VT,TCH,TSV,PC,RD
1  /C/XD,XDP,XDPP,XLS,XQ,XQP,XQPP,CID,CIQ,RLOAD,PI
1  /D/SCALE,PSVMIN,PSVMAX,VRMIN,VRMAX
```


DATA DER,Y/26*0.0/

C..... OPEN FILES TO OUTPUT DATA

```
C  OPEN(90,FILE='DQ.DAT',STATUS='NEW')
    OPEN(91,FILE='RBIABCDQ.DAT',STATUS='NEW')
    OPEN(92,FILE='RBY4.DAT',STATUS='NEW')
C  OPEN(92,FILE='Y16.DAT',STATUS='NEW')
C  OPEN(93,FILE='Y713.DAT',STATUS='NEW')
    OPEN(94,FILE='RBDFT.DAT',STATUS='NEW')
```

C..... INITIALIZE CONSTANTS

```
VT   = 1
VREF = 1.0+1./400.
TOTALT = 0.5
DELTAT = 1./3600.
T     = 0.0
WS    = 60.*2.*3.14159
N     = 60
XDPP  = 0.27
XQ    = 2.1
TE    = 0.8
KF    = 0.06
VRMAX = 5.0
VRMIN = 0.0
PSVMAX = 5.0
PSVMIN = 0.0
RS    = 0.0017
XLS   = 0.2
XQP   = 0.83
KE    = 1.0
TA    = 0.2
TCH   = 0.3
PC    = 20.0
H     = 3.2
XD    = 2.5
TDOP  = 9.6
TDOPP = 0.08
XQPP  = 0.29
SE    = 0.0
KA    = 400
TSV   = 0.2
P     = 2
XDP   = 0.39
```

```

TQOP = 1.7
TQOPP = 0.14
TF = 1.0
RD = 0.05
Y(4) = WS
PI = 4.*ATAN(1.0)
SCALE = 0.0
NDP = INT(TOTALT/DELTAT)

```

C..... INITIAL CONDITONS

```

Y(1) = 1.0
Y(2) = 0.0
Y(3) = 0.0
Y(4) = WS
Y(5) = 1.0
Y(6) = 1.0
Y(7) = 0.0
Y(8) = 0.0
Y(9) = 1.0
Y(10) = 0.06
Y(11) = 1.0
Y(12) = 0.0
Y(13) = 0.0
VD = 0.0
VQ = 1.0

```

C..... START MAIN PROGRAM LOOP

```

DO 100 K=0,NDP
  T = K*DELTAT

```

```

  CALL RUNGE(Y,T,DELTAT)
  CALL PHASORS(Y,T,MAGVDQ,PHASEVDQ,MAGIDQ,PHASEIDQ,IA,IB,IC)
  CALL DFT(N,K,IA,IB,IC,DFTMAG,DFTPH)

```

```

  IF (MOD(K,1).EQ.0) THEN

```

```

C      WRITE(90,102) T,CID,CIQ,VD,VQ
      WRITE(91,103) T,IA,IB,IC,MAGIDQ,PHASEIDQ
      WRITE(92,104) T,Y(4)
C      WRITE(92,103) T,Y(1),Y(2),Y(3),Y(4),Y(5),Y(6)
C      WRITE(93,103) T,Y(7),Y(8),Y(9),Y(10),Y(11),Y(12),Y(13)
      WRITE(94,101) T,DFTMAG,DFTPH
101  FORMAT(1X,F11.8,1X,F8.3,1X,F8.3,1X,F8.3,1X,F8.3)

```

```

102 FORMAT(1X,F11.8,1X,F8.3,1X,F8.3,1X,F8.3,1X,F8.3,1X,F8.3,1X,F8.3)
103 FORMAT(1X,F11.8,1X,F8.3,1X,F8.3,1X,F8.3,1X,F8.3,1X,F8.3,1X,F8.3,
1 ,1X,F8.3)
104 FORMAT(1X,F11.8,1X,F8.3)
      ENDIF

```

```

100 CONTINUE
      END

```

C....SUBROUTINE TO PERFORM INTEGRATION BY FOURTH ORDER RUNGE-KUTTA

```

      SUBROUTINE RUNGE(Y,T,DELTAT)
      DOUBLEPRECISION Y(13),DER(13),K1(13),K2(13),K3(13),K4(13),
1 N,P,IA,IB,IC,DELTAT,YTEMP(13),
1 MAGIDQ,PHASEIDQ,MAGVDQ,PHASEVDQ,WS,RS,VD,VQ,H,TDOPP,TQOP,R,
2 TA,TE,KE,SE,TF,KF,KA,VREF,VT,TCH,TSV,PC,RD,XD,XDP,XDPP,XLS,
3 XQ,XQP,XQPP,CID,CIQ,RLOAD,PI,PSVMIN,PSVMAX,VRMIN,VRMAX,T,
4 TEMP1D,TEMP2D,TEMP1Q,TEMP2Q

      COMMON /A/WS,RS,VD,VQ,H,TDOP,TDOPP,TQOP,TQOPP,R
1 /B/TA,TE,KE,SE,TF,KF,KA,VREF,VT,TCH,TSV,PC,RD
1 /C/XD,XDP,XDPP,XLS,XQ,XQP,XQPP,CID,CIQ,RLOAD,PI

      CALL DERIVE(Y,T,K1)
      DO 10 I=1,13
          K1(I)=DELTAT*K1(I)
          YTEMP(I)=Y(I)+K1(I)/2.
          YTEMP(1)=VQ+RS*CIQ
          YTEMP(2)=-VD-RS*CID
          CALL CURRENT(YTEMP,T)
          CALL LIMITS(YTEMP,T)
10 CONTINUE
          CALL DERIVE(YTEMP,T+DELTAT/2.,K2)
          DO 20 I=1,13
              K2(I)=DELTAT*K2(I)
              YTEMP(I)=Y(I)+K2(I)/2.
              YTEMP(1)=VQ+RS*CIQ
              YTEMP(2)=-VD-RS*CID
              CALL CURRENT(YTEMP,T)
              CALL LIMITS(YTEMP,T)
20 CONTINUE
          CALL DERIVE(YTEMP,T+DELTAT/2.,K3)
          DO 30 I=1,13
              K3(I)=DELTAT*K3(I)

```

```

        YTEMP(I)=Y(I)+K3(I)
        YTEMP(1)=VQ+RS*CIQ
        YTEMP(2)=-VD-RS*CID
        CALL CURRENT(YTEMP,T)
        CALL LIMITS(YTEMP,T)
30  CONTINUE
        CALL DERIVE(YTEMP,T+DELTAT,K4)
        DO 40 I=1,13
            K4(I)=DELTAT*K4(I)
            Y(I)=Y(I)+(1./6.)*(K1(I)+2.*K2(I)+2.*K3(I)+K4(I))
            Y(1)=VQ+RS*CIQ
            Y(2)=-VD-RS*CID
            CALL CURRENT(Y,T)
            CALL LIMITS(Y,T)
40  CONTINUE
        RETURN
        END

C.....  THE STATES
C.....  Y(1)=PD      Y(6)=P1D      Y(11)=VR
C.....  Y(2)=PQ      Y(7)=EDP      Y(12)=TM
C.....  Y(3)=D       Y(8)=P2Q      Y(13)=PSV
C.....  Y(4)=W       Y(9)=EFD
C.....  Y(5)=EQP     Y(10)=RF

C.....  SUBROUTINE TO CALCULATE THE DERIVATIVES OF THE STATES

        SUBROUTINE DERIVE(Y,T,DER)
        DOUBLEPRECISION DER(13),Y(13),N,P,IA,IB,IC,
1  MAGIDQ,PHASEIDQ,MAGVDQ,PHASEVDQ,WS,RS,VD,VQ,H,TDOPP,TQOP,R,
2  TA,TE,KE,SE,TF,KF,KA,VREF,VT,TCH,TSV,PC,RD,XD,XDP,XDPP,XLS,
3  XQ,XQP,XQPP,CID,CIQ,RLOAD,PI,PSVMIN,PSVMAX,VRMIN,VRMAX,T,
4  T1,T2,X1,X2,X3,X4

        COMMON /A/WS,RS,VD,VQ,H,TDOP,TDOPP,TQOP,TQOPP,R
1  /B/TA,TE,KE,SE,TF,KF,KA,VREF,VT,TCH,TSV,PC,RD
1  /C/XD,XDP,XDPP,XLS,XQ,XQP,XQPP,CID,CIQ,RLOAD,PI

        T1  = (XDP-XDPP)/((XDP-XLS)**2)
        T2  = (XQP-XQPP)/((XQP-XLS)**2)
        X1  = XD-XDP
        X2  = XDP-XLS
        X3  = XQ-XQP
        X4  = XQP-XLS
        DER(1) = WS*(RS*CID+Y(4)*(Y(2)/WS)+VD)

```



```

DER(2) = WS*(RS*CIQ-Y(4)*(Y(1)/WS)+VQ)
DER(3) = Y(4)-WS
DER(4) = (WS/(2.*H))*(Y(12)-Y(1)*CIQ+Y(2)*CID)
DER(5) = (-Y(5)-X1*(CID-T1*(Y(6)+X2*CID-Y(5)))+Y(9))/TDOP
DER(6) = (-Y(6)+Y(5)-(X2)*CID)/TDOPP
DER(7) = (-Y(7)+X3*(CIQ-T2*(Y(8)+X4*CIQ+Y(7))))/TQOP
DER(8) = (1./TQOPP)*(-Y(8)-Y(7)-(X4)*CIQ)
DER(9) = (1./TE)*(-(KE+SE)*Y(9)+Y(11))
DER(10) = (1./TF)*(-Y(10)+(KF/TF)*Y(9))
DER(11) = (1./TA)*(-Y(11)+KA*Y(10)-(KA*KF/TF)*Y(9)+KA*(VREF-VT))
DER(12) = (1./TCH)*(-Y(12)+Y(13))
DER(13) = (1./TSV)*(-Y(13)+PC-(1./RD)*(Y(4)/WS))
RETURN
END

```

C..... SUBROUTINE TO CALCULATE CURRENT, VOLTAGE AND SET LOAD

SUBROUTINE CURRENT(Y,T)

```

DOUBLEPRECISION DER(13),Y(13),N,P,IA,IB,IC,
1 MAGIDQ,PHASEIDQ,MAGVDQ,PHASEVDQ,WS,RS,VD,VQ,H,TDOPP,TQOP,R,
2 TA,TE,KE,SE,TF,KF,KA,VREF,VT,TCH,TSV,PC,RD,XD,XDP,XDPP,XLS,
3 XQ,XQP,XQPP,CID,CIQ,RLOAD,PI,PSVMIN,PSVMAX,VRMIN,VRMAX,T,
4 TEMP1D,TEMP2D,TEMP1Q,TEMP2Q

```

```

COMMON /A/WS,RS,VD,VQ,H,TDOP,TDOPP,TQOP,TQOPP,R
1 /B/TA,TE,KE,SE,TF,KF,KA,VREF,VT,TCH,TSV,PC,RD
1 /C/XD,XDP,XDPP,XLS,XQ,XQP,XQPP,CID,CIQ,RLOAD,PI

```

```

TEMP1D = (XDPP-XLS)/(XDP-XLS)
TEMP2D = (XDP-XDPP)/(XDP-XLS)
TEMP1Q = (XQPP-XLS)/(XQP-XLS)
TEMP2Q = (XQP-XQPP)/(XQP-XLS)
A11 = 1.0
A12 = RS/(XDPP+RLOAD)
A21 = -RS/(XQPP-RLOAD)
A22 = 1.0
DET = 1/(A11*A22-A21*A12)
B1 = (1/(XDPP+RLOAD))*(TEMP1D*Y(5)+TEMP2D*Y(6))
B2 = (1/(XQPP-RLOAD))*(-TEMP1Q*Y(7)+TEMP2Q*Y(8))
IF (T.LT.0.05) THEN
    CID = 0.0
    CIQ = 0.0
ENDIF
IF ((T.GE.0.05).AND.(T.LT.0.15)) THEN

```

```

        RLOAD=0.01
        CID = -DET*(-A22*B1+A12*B2)
        CIQ = -DET*(A21*B1-A11*B2)
        VD = CID*RLOAD
        VQ = CIQ*RLOAD
        VT = SQRT((VD**2)+(VQ**2))
    ENDIF
    IF (T.GE.0.15) THEN
        RLOAD=2.0
        CID = -DET*(-A22*B1+A12*B2)
        CIQ = -DET*(A21*B1-A11*B2)
        VD = CID*RLOAD
        VQ = CIQ*RLOAD
        VT = SQRT((VD**2)+(VQ**2))
    ENDIF
    RETURN
END

```

C..... SUBROUTINE TO CALCULATE PHASORS AND ABC CURRENTS

```

SUBROUTINE
PHASORS(Y,T,MAGVDQ,PHASEVDQ,MAGIDQ,PHASEIDQ,IA,IB,IC)
    DOUBLEPRECISION DER(13),Y(13),N,P,IA,IB,IC,
    1 MAGIDQ,PHASEIDQ,MAGVDQ,PHASEVDQ,WS,RS,VD,VQ,H,TDOPP,TQOP,R,
    2 TA,TE,KE,SE,TF,KF,KA,VREF,VT,TCH,TSV,PC,RD,XD,XDP,XDPP,XLS,
    3 XQ,XQP,XQPP,CID,CIQ,RLOAD,PI,PSVMIN,PSVMAX,VRMIN,VRMAX,T,
    4 TEMPVD,TEMPID

```

```

    COMMON /A/WS,RS,VD,VQ,H,TDOP,TDOPP,TQOP,TQOPP,R
    1 /B/TA,TE,KE,SE,TF,KF,KA,VREF,VT,TCH,TSV,PC,RD
    1 /C/XD,XDP,XDPP,XLS,XQ,XQP,XQPP,CID,CIQ,RLOAD,PI
    1 /D/SCALE,PSVMIN,PSVMAX,VRMIN,VRMAX

```

```

        MAGVDQ = VT
        IF (ABS(VD).LT.0.00001) THEN
            TEMPVD=0.00001
        ELSE
            TEMPVD=VD
        ENDIF
        PHASEVDQ=(ATAN2(VQ,TEMPVD))+Y(3)-PI/2.
        MAGIDQ=SQRT((CID**2)+(CIQ**2))

        TEMPID=CID

```

```

IF (ABS(TEMPID).LT.0.00001) THEN
    TEMPID=0.00001
ELSE
    TEMPID=CID
ENDIF
PHASEIDQ=(ATAN2(CIQ,TEMPID))+Y(3)-(PI/2.)+(SCALE*2.*PI)
IF (PHASEIDQ.LT.-PI) THEN
    SCALE=SCALE+1.0
ELSEIF (PHASEIDQ.GT.PI) THEN
    SCALE=SCALE-1.0
ENDIF

PHASEIDQ=(ATAN2(CIQ,TEMPID))+Y(3)-(PI/2.)+(SCALE*2.*PI)

IF (T.LT.0.05) THEN
    PHASEIDQ=0.0
ENDIF

IA=SQRT(2.)*MAGIDQ*COS(WS*T+PHASEIDQ)
IB=SQRT(2.)*MAGIDQ*COS(WS*T+PHASEIDQ-2.*PI/3.)
IC=SQRT(2.)*MAGIDQ*COS(WS*T+PHASEIDQ+2.*PI/3.)

RETURN
END

```

C..... SUBROUTINE TO CALCULATE DFT'S

```

SUBROUTINE DFT(N,K,IA,IB,IC,DFTMAG,DFTPH)
DOUBLEPRECISION DFTMAG,DFTPH,IA,IB,IC,PI
COMPLEX PHIA,PHIB,PHIC,ALPHA,X1,SPHIA(60),SPHIB(60),SPHIC(60),
1 SPHIA2(60),SPHIB2(60),SPHIC2(60)

PI=4.*ATAN(1.0)
KCY=KCY+1
ALPHA=CMPLX(COS(2.*PI/3.),SIN(2.*PI/3.))

RIA=(SQRT(2.)/N)*IA*COS(2.*KCY*PI/N)
XIA=(SQRT(2.)/N)*IA*(-SIN(2.*KCY*PI/N))
SPHIA2(KCY)=CMPLX(RIA,XIA)
PHIA=PHIA+CMPLX(RIA,XIA)
APHIA=CABS(PHIA)
RIA=REAL(PHIA)
XIA=AIMAG(PHIA)
TEMPRIA=RIA

```



```
IF (ABS(TEMPRIA).LT.0.00001) THEN
```

```
    TEMPRIA=0.00001
```

```
ELSE
```

```
    TEMPRIA=RIA
```

```
ENDIF
```

```
PPHIA=ATAN2(XIA,TEMPRIA)
```

```
RIB=(SQRT(2.)/N)*IB*COS(-2.*KCY*PI/N)
```

```
XIB=(SQRT(2.)/N)*IB*(-SIN(2.*KCY*PI/N))
```

```
SPHIB2(KCY)=CMPLX(RIB,XIB)
```

```
PHIB=PHIB+CMPLX(RIB,XIB)
```

```
APHIB=CABS(PHIB)
```

```
RIB=REAL(PHIB)
```

```
XIB=AIMAG(PHIB)
```

```
TEMPRIB=RIB
```

```
IF (ABS(TEMPRIB).LT.0.00001) THEN
```

```
    TEMPRIB=0.00001
```

```
ELSE
```

```
    TEMPRIB=RIB
```

```
ENDIF
```

```
PPHIB=ATAN2(XIB,TEMPRIB)
```

```
RIC=(SQRT(2.)/N)*IC*COS(2.*KCY*PI/N)
```

```
XIC=(SQRT(2.)/N)*IC*(-SIN(2.*KCY*PI/N))
```

```
SPHIC2(KCY)=CMPLX(RIC,XIC)
```

```
PHIC=PHIC+CMPLX(RIC,XIC)
```

```
APHIC=CABS(PHIC)
```

```
RIC=REAL(PHIC)
```

```
XIC=AIMAG(PHIC)
```

```
TEMPRIC=RIC
```

```
IF (ABS(TEMPRIC).LT.0.00001) THEN
```

```
    TEMPRIC=0.00001
```

```
ELSE
```

```
    TEMPRIC=RIC
```

```
ENDIF
```

```
PPHIC=ATAN2(XIC,TEMPRIC)
```

```
X1=(1./3.)*(PHIA+ALPHA*PHIB+ALPHA*ALPHA*PHIC)
```

```
C NEG    X1=(1./3.)*(PHIA+ALPHA*ALPHA*PHIB+ALPHA*PHIC)
```

```
C ZERO    X1=(1./3.)*(PHIA+PHIB+PHIC)
```

```
AX1=CABS(X1)
```

```
TEMPX1=REAL(X1)
```

```
IF (ABS(TEMPX1).LT.0.00001) THEN
```

```
    TEMPX1=0.00001
```

```
ELSE
```



```

        TEMPX1=REAL(X1)
ENDIF
PX1=ATAN2(AIMAG(X1),TEMPX1)

DFTMAG=AX1
DFTPH=PX1

IF (KCY.EQ.N) THEN
        DO 400 I=1,N
                SPHIA(I)=SPHIA2(I)
                SPHIB(I)=SPHIB2(I)
                SPHIC(I)=SPHIC2(I)
                KCY=0
400    CONTINUE
ENDIF

IF (K.GE.N) THEN
        PHIA=PHIA-SPHIA(1)
        PHIB=PHIB-SPHIB(1)
        PHIC=PHIC-SPHIC(1)
        DO 401 J=2,N
                SPHIA(J-1)=SPHIA(J)
                SPHIB(J-1)=SPHIB(J)
                SPHIC(J-1)=SPHIC(J)
401    CONTINUE
ENDIF

RETURN
END

SUBROUTINE LIMITS(Y,T)
DOUBLEPRECISION Y(13),T,VRMIN,VRMAX,PSVMIN,PSVMAX

COMMON /D/SCALE,PSVMIN,PSVMAX,VRMIN,VRMAX

        IF (Y(11).LT.VRMIN) THEN
                Y(11)=VRMIN
        ENDIF
        IF (Y(11).GT.VRMAX) THEN
                Y(11)=VRMAX
        ENDIF
        IF (Y(13).LT.PSVMIN) THEN
                Y(13)=PSVMIN
        ENDIF
        IF (Y(13).GT.PSVMAX) THEN

```

```

                Y(13)=PSVMAX
            ENDIF
        RETURN
    END

```

A.3 Half-Cycle Phasor Subroutine

C..... SUBROUTINE TO CALCULATE DFT'S

```

SUBROUTINE DFT(N,K,IA,IB,IC,DFTMAG,DFTPH)
DOUBLEPRECISION DFTMAG,DFTPH,IA,IB,IC,PI
COMPLEX PHIA,PHIB,PHIC,ALPHA,X1,SPHIA(60),SPHIB(60),SPHIC(60),
1 SPHIA2(60),SPHIB2(60),SPHIC2(60)

```

```

    PI=4.*ATAN(1.0)
    KCY=KCYN+1
    ALPHA=CMPLX(COS(2.*PI/3.),SIN(2.*PI/3.))

```

```

    RIA=(SQRT(2.)/N)*IA*COS(2.*K*PI/N)
    XIA=(SQRT(2.)/N)*IA*(-SIN(2.*K*PI/N))
    SPHIA2(KCY)=CMPLX(RIA,XIA)
    PHIA=PHIA+CMPLX(RIA,XIA)
    APHIA=CABS(PHIA)
    RIA=REAL(PHIA)
    XIA=AIMAG(PHIA)
    TEMPRIA=RIA
    IF (ABS(TEMPRIA).LT.0.00001) THEN
        TEMPRIA=0.00001
    ELSE
        TEMPRIA=RIA
    ENDIF
    PPHIA=ATAN2(XIA,TEMPRIA)

```

```

    RIB=(SQRT(2.)/N)*IB*COS(2.*K*PI/N)
    XIB=(SQRT(2.)/N)*IB*(-SIN(2.*K*PI/N))
    SPHIB2(KCY)=CMPLX(RIB,XIB)
    PHIB=PHIB+CMPLX(RIB,XIB)
    APHIB=CABS(PHIB)
    RIB=REAL(PHIB)
    XIB=AIMAG(PHIB)
    TEMPTRIB=RIB
    IF (ABS(TEMPTRIB).LT.0.00001) THEN
        TEMPTRIB=0.00001
    ELSE

```

```

        TEMPRIB=RIB
    ENDIF
    PPHIB=ATAN2(XIB,TEMPRIB)

    RIC=(SQRT(2.)/N)*IC*COS(2.*K*PI/N)
    XIC=(SQRT(2.)/N)*IC*(-SIN(2.*K*PI/N))
    SPHIC2(KCY)=CMPLX(RIC,XIC)
    PHIC=PHIC+CMPLX(RIC,XIC)
    APHIC=CABS(PHIC)
    RIC=REAL(PHIC)
    XIC=AIMAG(PHIC)
    TEMPRIC=RIC
    IF (ABS(TEMPRIC).LT.0.00001) THEN
        TEMPRIC=0.00001
    ELSE
        TEMPRIC=RIC
    ENDIF
    PPHIC=ATAN2(XIC,TEMPRIC)

    X1=2.*(1./3.)*(PHIA+ALPHA*PHIB+ALPHA*ALPHA*PHIC)
    C NEG    X1=(1./3.)*(PHIA+ALPHA*ALPHA*PHIB+ALPHA*PHIC)
    C ZERO    X1=(1./3.)*(PHIA+PHIB+PHIC)
    AX1=CABS(X1)
    TEMPX1=REAL(X1)
    IF (ABS(TEMPX1).LT.0.00001) THEN
        TEMPX1=0.00001
    ELSE
        TEMPX1=REAL(X1)
    ENDIF
    PX1=ATAN2(AIMAG(X1),TEMPX1)

    DFTMAG=AX1
    DFTPH=PX1

    IF (KCX.EQ.INT(N/2.)) THEN
        DO 400 I=1,INT(N/2.)
            SPHIA(I)=SPHIA2(I)
            SPHIB(I)=SPHIB2(I)
            SPHIC(I)=SPHIC2(I)
            KCY=0
400    CONTINUE
    ENDIF

    IF (K.GE.INT(N/2.)) THEN
        PHIA=PHIA-SPHIA(1)

```

```
      PHIB=PHIB-SPHIB(1)
      PHIC=PHIC-SPHIC(1)
      DO 401 J=2,INT(N/2.)
          SPHIA(J-1)=SPHIA(J)
          SPHIB(J-1)=SPHIB(J)
          SPHIC(J-1)=SPHIC(J)
401      CONTINUE
      ENDIF

      RETURN
      END
```